Stochastic Clearing Systems with Markovian Inputs: Performance Evaluation and Optimal Policies

by

Qishu Cai

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Management Sciences

Waterloo, Ontario, Canada, 2015

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis studies the stochastic clearing systems which are characterized by a non-decreasing stochastic input process $\{Y(t), t \geq 0\}$, where $Y(t)$ is the cumulative quantity entering the system in $[0, t]$, and an output mechanism that intermittently and instantaneously clears the system. Examples of such systems can be found in shipment consolidation, inventory backlog, lot sizing, shuttle bus dispatch, bulk service queues, and other stochastic service and storage systems. In our model, the input process is governed by an underlying discrete-time Markov chain such that, the distribution of the input in any given period depends on the underlying state in that period. The outstanding inputs in the system are recorded in strings to keep track of the ages, i.e., the time elapsed since their arrival, of each input. The decision of when to clear the system depends on a "clearing policy" which itself depends on the input quantities, ages, and the underlying state. Clearing the system will incur a fixed cost and a variable cost depending on the quantities cleared; a penalty is charged to the outstanding inputs in every period, and such penalty is non-decreasing in both the quantities and the ages of the inputs. We model the system as a tree structured Markov chain with Markovian input processes and evaluate the clearing policies with respect to the expected total costs over a finite horizon, the expected total discounted cost over an infinite horizon, as well as the expected average total cost per period over an infinite horizon. Relying on theories of Markov Decision Processes and stochastic dynamic programming, we then proceed to show some properties unique to the optimal clearing policies, and prove that a state-dependent threshold policy can be optimal under special conditions. We develop algorithms or heuristics to evaluate a given clearing policy and find the optimal clearing policy. We also use Matrix Analytic Methods to evaluate a given clearing policy and develop an efficient heuristic to find near-optimal clearing policies. Finally, we conduct extensive numerical analyses to verify the correctness, complexity, and optimality gap of our algorithms and heuristics. Our numerical examples successfully demonstrate the analytical results we proved.

Acknowledgements

I would like to express gratitude to my PhD advisors, Dr. James H. Bookbinder and Dr. Qi-Ming He, for their guidance and support over the course of my graduate studies. Their passion and dedication to the field of operation research and management science have been a tremendous influence on me. I would like to thank Dr. Tim Huh, who kindly offered me valuable suggestions during the CORS conference in 2012, which then became the inspiration for some key ideas in my PhD research. Lastly, I would like to show my appreciation to the members of my PhD committee, Dr. Elizabeth Jewkes, Dr. Steve Drekic, and Dr. Hossein Abouee Mehrizi, for their comments and suggestions on how to improve my research and writing of this thesis.

Dedication

This is dedicated to my parents who have sacrificed so much to give me the best education possible and always encourage me to pursue my dreams.

Table of Contents

List of Tables **xi**

List of Figures **xiii**

1 Introduction **1**

1.1 Model Introduction 1

1.1.1 Model Overview 2

1.1.2 Shuttle Bus Dispatch Example 4

1.1.3 Restaurant Delivery Example 5

1.1.4 Less-Than-Truckload Carrier Example 6

1.1.5 Luxury Car Dealership Example 7

1.2 Literature Review 8

1.2.1 Stochastic Clearing Systems 8

1.2.2 Shipment Consolidation Problems 10

1.2.3 Inventory Theories 14

1.2.4 Other Related Problems and Theories 17

1.2.5 Potential Research Problems 19

1.3 Methodology 20
1.3.1 Markov Decision Processes and Stochastic Dynamic Programming . 20
1.3.2 Markov Chain and Matrix Analytic Methods 23
1.4 Thesis Layout 25

2 Model Formulation **27**
2.1 Problem Definition and Notation 27
2.1.1 Decision Epochs, Planning Horizon, and Periods 27
2.1.2 Input Processes 29
2.1.3 System Contents 33
2.1.4 Action Set, Decision Rules, and Clearing Policies 37
2.2 Two Markov Chains 40
2.2.1 System State Spaces 40
2.2.2 Pre-Clearing System State Process 46
2.2.3 Post-Clearing System State Process 48
2.3 Cost Structure 51
2.3.1 Delay Penalty Cost 52
2.3.2 Clearing Costs 53
2.3.3 Objective Cost Functions 54

3 Expected Total Cost Model over Finite Horizon **56**
3.1 Policy Evaluation 57
3.2 Optimality Equations 61

3.3 Characterizing the Optimal Policies . . . 65
3.3.1 Effects of Initial System State . . . 66
3.3.2 Policy Boundaries . . . 72
3.4 State-Dependent Threshold Policies . . . 75
3.4.1 Special Delay Penalty Functions . . . 75
3.4.2 Characteristics of State-Dependent Threshold Policies . . . 78
3.5 Computing the Optimal Policy Parameters . . . 80
3.6 Summary of Results . . . 86

4 Expected Total Discounted Cost Model over Infinite Horizon 88

4.1 Policy Evaluation . . . 89
4.2 Existence of Optimal Policies . . . 93
4.3 Characterizing the Optimal Policies . . . 98
4.4 Computing the Optimal Policy Parameters . . . 101
4.4.1 Value Iteration Algorithm . . . 102
4.4.2 Policy Iteration Algorithm . . . 106
4.5 Summary of Results . . . 110

5 Expected Average Cost Model over Infinite Horizon (MDP) 112

5.1 Policy Evaluation . . . 113
5.1.1 Markov Reward Process . . . 113
5.1.2 Policy Evaluation Equations . . . 116
5.2 Optimality Equation . . . 119

5.3 Computing the Optimal Policy Parameters . . . 122
5.3.1 Value Iteration Algorithm . . . 122
5.3.2 Policy Iteration Algorithm . . . 124
5.4 Summary of Results . . . 127

6 Expected Average Cost Model over Infinite Horizon (MAM) 129

6.1 Policy Evaluation . . . 130
6.1.1 Stationary Distribution . . . 130
6.1.2 Clearing Cycles . . . 137
6.1.3 Long-Run Performance Measures . . . 138
6.1.4 Expected Average Cost per Period . . . 143
6.2 Computing the Optimal Policies . . . 146
6.2.1 General Properties . . . 146
6.2.2 State-Independent Threshold Policies . . . 148
6.3 Summary of Results . . . 154

7 Concluding Remarks 156

7.1 Thesis Summary . . . 156
7.2 Future Research . . . 158
7.2.1 Optimality of State-Dependent Threshold Clearing Policy . . . 158
7.2.2 Other Forms of Cost Structures . . . 159
7.2.3 Impact of Input Process . . . 159
7.2.4 Partially-Observable or Unobservable Underlying States . . . 160
7.2.5 Continuous Time and/or Continuous Quantity Models . . . 161

APPENDICES 162

A Linear Programming Formulation for Total Discounted Cost Model 163

B Linear Programming Formulation for Average Total Cost Model 166

C Complexity Studies 169

References 172

List of Tables

3.1 Summary of Expected Total Costs for Example 3.1.1 60

3.2 Summary of Optimal Clearing Rule in Period 3 for Example 3.5.1 83

3.3 Summary of Optimal Clearing Rule in Period 2 for Example 3.5.1 84

3.4 Summary of Optimal Clearing Rule in Period 1 for Example 3.5.1 84

3.5 Summary of Optimal Clearing Rule in Period 3 for Example 3.5.2 85

3.6 Summary of Optimal Clearing Rule in Period 2 for Example 3.5.2 85

3.7 Summary of Optimal Clearing Rule in Period 1 for Example 3.5.2 86

4.1 Summary of Expected Total Discounted Costs for Example 4.1.1 92

4.2 Summary of Optimal Clearing Rule for Example 4.4.1 109

4.3 Summary of Optimal Clearing Rule in Period 1 for Example 4.4.2 110

5.1 Summary of Expected Average Costs for Example 5.1.1 118

5.2 Summary of Optimal Clearing Rule for Example 5.3.1 126

5.3 Summary of Optimal Clearing Rule in Period 1 for Example 5.3.2 127

6.1 Summary of Expected Average Costs for Example 6.1.1 145

C.1 Complexity Summary 169

C.2 BFT Procedure I Time Complexity Tests 170

C.3 BFT Procedure I Memory Complexity Tests 171

List of Figures

2.1 A sample time line of the stochastic clearing process 28

2.2 A sample tree of Φ with $Q = 2$ 35

2.3 A sample tree of $\Phi_{(r_t)}$ with $Q = 2$ 43

6.1 Costs of Threshold Policies in Example 6.2.1 150

C.1 BFT Procedure I Complexity Tests 171

Chapter 1

Introduction

Stochastic clearing systems are a type of random input-output system with many practical applications. These systems receive and accumulate inputs of random quantities over random time intervals until certain predetermined criteria are met; then, there is a "clearing" of some or all of these inputs instantaneously. Examples of stochastic clearing systems include transit service for riders waiting at bus stops, delivery services, inventory management systems, dam control systems, and bulk service queues. In each case, the clearing of some or all inputs means "servicing" or "processing" previously arrived (waiting) inputs.

The main purpose of this type of system is to gain economies of scale by clearing multiple inputs together. Because of the random nature of the input process, however, the criterion for clearing must be selected carefully to achieve the desired result. Our objectives in studying stochastic clearing systems focus on evaluating the performance of an existing system, and optimizing the system control strategy.

1.1 Model Introduction

Stochastic clearing systems can be studied via a variety of stochastic models under different modelling assumptions. There are several key components found across different models.

1.1.1 Model Overview

Similar to inventory systems, stochastic clearing systems can be reviewed and managed periodically or continuously. In some practical applications, the system administrator makes clearing decisions only at discrete time epochs, but in other cases, the system state is constantly monitored and clearing takes place as soon as a clearing criterion is satisfied. Therefore, we can model the input interarrival times as discrete or continuous random variables.

Depending on the nature of the inputs and how they are accounted for, input quantities can also be discrete or continuous. For example, passengers arriving at a bus stop obviously constitute discrete input, but the flow of water into a dam is measured as a continuous random variable. System inputs may be classified into distinctive types because they incur different costs, need to be processed under different priorities, or require special treatment in the system.

Together, the input interarrival times and quantities are modelled by a stochastic input process. Some commonly used input processes include the Poisson process, compound Poisson process, Markovian arrival process, and batch Markovian arrival process. In this thesis, we focus primarily on the Markovian and batch Markovian arrival processes, in which the interarrival times and input quantities may be correlated and affected by an underlying state-of-the-world variable. These types of input processes are particularly potent at capturing the correlations between different inputs and the variations of input rate over time.

The next important modelling decision is how to record the accumulated input quantities. In the existing literature of stochastic clearing systems, the total cumulative quantity is usually the only variable used to keep track of the system state. In that case, there is a lack of information about the amount of time which an input has remained in the system, hence a recently acquired input is perceived as equally "urgent" as another input that was

received a long time ago. It is easy to see the flaw in this assumption because in many stochastic clearing systems, time spent in the system is an important factor in deciding when an input must be cleared.

In our models, the quantities and times in the system are recorded separately for individual inputs. This set of information will help us make better clearing decisions with respect to the varying levels of urgency of different inputs. Note that the added dimensions of the system state variable increase the time and memory complexities of our algorithms, but the improved accuracy and effectiveness of our models can justify the added time and cost of computation.

After deciding on how to keep track of the system state, we need to find a standard way to represent the clearing criteria. Since the clearing decision is contingent upon time and system state, we shall model the clearing decision as a function of time and system state. If the clearing criterion is stationary over time, that function is independent of time. We shall assume that clearing is instantaneous so that no input may be received during the act of clearing.

In a conventional stochastic clearing system, once it is decided to clear the system, all accumulated input quantities must be cleared and the system immediately becomes empty. In our study, for the sake of better understanding the process, we shall allow some input quantities to be left behind. Therefore, the output of the clearing function is in fact the system state after clearing.

Lastly, we need to specify the measures that are going to be used to assess the performance of a particular clearing policy. These measures are usually in terms of costs associated with the process such as the cost of clearing the system and the cost of holding inputs in the system. There may also be some other non-financial metrics such as the average time an input stays in the system until clearing and the average quantity cleared each time. In terms of cost, we can calculate the expected total cost of the system in the

short term, the expected discounted total cost if the process is stretched over an extended period of time, or the expected average total cost per period in the long run.

In the following sections, we describe several examples of stochastic clearing systems and demonstrate how they fit into our modelling framework.

1.1.2 Shuttle Bus Dispatch Example

The shuttle bus dispatch problem is one of the classical stochastic clearing problems. Ignall and Kolesar [47] and Deb [26] were among the first to study this problem. Subsequently, the shuttle bus disptach problem is categorized as the batch service queueing problem by Papdaki and Powell [63, 64].

Consider a shuttle bus operation at an airport or train station; passengers arrive individually or in groups to take the shuttle bus heading for a common destination. Usually, a vehicle will depart immediately after it is filled with passengers, but quite often, the vehicle will be dispatched even if it is not full. This is because sometimes it is highly unlikely to get any other passengers within a reasonable amount of time, and the passengers on board may get impatient if they are kept waiting for too long. Assuming there is an abundance of vehicles and drivers, the important question to address is under what conditions a vehicle should be dispatched.

The passenger arrival process can be effectively modelled by a batch Markovian arrival process, not only because the passengers can arrive in groups, but also because there are some correlations among arrivals. In an airport or train station, shuttle bus passengers arrive on a landed plane or docked train. Since the planes and trains arrive periodically, after each plane or train arrival, there is likely a burst of passenger arrivals. In addition to that, the plane and train arrivals may not be evenly scheduled throughout the day. For example, in many airports, there are more arrivals in the evenings than in the mornings. Therefore, there may be some seasonality effects on the passenger arrival process. The

batch Markovian arrival process is proven to be effective for modelling this kind of situation.

In the shuttle bus operation, it is important to recognize the different needs of passengers. For instance, business travellers may require more urgent departure while leisure travellers can afford to wait a bit longer; group travellers have more leverage than individual passengers in their demand for early departure; and walk-in customers are more likely to leave with other services than prepaid customers. Keeping track of individual groups of passengers and the passenger types thus become crucial in managing the dispatches. Note that due to the capacity of the shuttle bus, the sizes of the passenger groups are limited, thus reducing the complexity of our model.

1.1.3 Restaurant Delivery Example

Food delivery service is another example of a stochastic clearing system. The practice is particularly prevalent in pizza restaurants and other fast food services, where small orders of varying amounts need to be sent to the customers frequently. It is a common practice to combine several nearby orders and send them by a single delivery person. In some cases, the consolidation is necessary because there are not enough resources, i.e., manpower and vehicles.

A batch Markovian arrival process is again a good choice for modelling the input process because, firstly, each delivery order may vary in value or size depending on the content of the order, and secondly, there is a strong seasonality effect throughout the day which affects the frequency of orders. Capturing these effects helps ensure that our model is an accurate rendering of the business.

The delivery waiting time is an important issue for food delivery services. It affects the quality of the food as it directly correlates to freshness. It also affects the amount of tips a delivery person can receive on an order. To promote the delivery business, some restaurants even agree to waive the cost of the entire order if delivery takes longer than

the promised time. Therefore, the clearing policy determines when and how many orders a delivery person must take on a trip, and it should maximize the service level with the limited delivery resources.

1.1.4 Less-Than-Truckload Carrier Example

On a larger scale, stochastic clearing systems fall into the category of the shipment consolidation problems in transportation and logistics management. One example is the less-than-truckload (LTL) carrier operation. When a company or an individual needs to ship something over a long distance but does not have enough quantity to occupy an entire container, they can hire an LTL carrier. Having received separate delivery orders of relatively small sizes, the LTL carrier will try to combine shipments with common destinations into a single larger load because a fixed cost is incurred upon dispatch, regardless of the content of the load.

LTL carriers may also incur a holding cost for physically storing the shipments, and delay penalties for not being able to deliver on time. Depending on the content of the shipments and the waiting time, the LTL carrier needs to decide whether it is cost effective to ship all orders on hand or continue to wait for more orders to come in. The delay penalty may be specified as a predetermined amount in the delivery contract, or in the form of an opportunity cost of lost future business due to unsatisfactory services.

In shipment consolidation, the size of a shipment can be given in terms of its weight or volume. There may also be a transportation cost proportional to the quantity of the consolidated load. Sometimes, there may even be a quantity discount for shipping more quantity at a time. Although weight and volume are normally continuous measures, in practice, many shippers choose to discretize them to appropriate units. In the shipment consolidation problem, other service measures such as the average delay of delivery and the average size of consolidated load are also important measures for both the shipper and

the LTL carrier.

1.1.5 Luxury Car Dealership Example

The last example of stochastic clearing systems we will discuss here is a car dealership, in particular, a luxury car dealership. We are interested in this example because unlike the previous examples, the clearing process here is often stretched over an extended period of time, i.e., weeks or even months, so the associated costs may be accounted for as discounted costs.

In a luxury car dealership, there is rarely any inventory except for the cars on display because this type of products is highly customized and are sold on demand. If a buyer is interested in a particular vehicle, he or she will place an order with the dealer and wait for the dealer to procure the car from the manufacturer.

Note that the geographic location (often across continents), rarity (assembled upon receiving the customized order), and high value of the product (can be as high as hundreds of thousands of dollars) often results in large procurement, transportation and insurance fees. Sometimes, the dealer will try to reduce these costs by procuring and shipping several cars together. This is essentially an inventory backlog problem.

Studies have shown that the demand for luxury cars is highly dependent on the economic environment, for example, the stock market performance. Therefore, that can be considered as an underlying state-of-the-world which affects the input process. Although buyers are usually willing to wait to get their desired cars, competitors may try to steal the customer away by offering to deliver the car early. Also note that significant inventory holding costs are charged to the dealer when a car is being transported to the dealership. Therefore, our stochastic clearing model can be used to study this problem.

Now that we have described the general model components and practical examples, we will review the literature on stochastic clearing systems and other related problems in the

next section.

1.2 Literature Review

Theories of stochastic clearing systems have long been studied by both mathematicians and practitioners. We shall first review some of the prominent works in the general framework of stochastic clearing problems, then look at some well-known applications in shipment consolidation, inventory theory, and several other problems.

1.2.1 Stochastic Clearing Systems

Stochastic clearing systems were first systematically modelled and studied by Stidham in [77]. A number of examples found in different areas such as queueing, inventory, stochastic service, and storage system, were described. The model is characterized by a non-decreasing stochastic input process and an instantaneous clearing that resets the system into the empty state. In that paper, the process was considered as regenerative and an explicit expression for the stationary distribution of the quantity in the system was derived. An important observation was made on the stationary distribution not being uniform for genuinely stochastic models, which suggests that the stochastic nature of the input process directly affects the optimal clearing decsisions.

In his subsequent paper [78], Stidham generalized the clearing operations to allow restoring the net quantity in the system to a level different than zero, i.e., not everything must be cleared at once. The costs associated with the process are identified as the fixed clearing cost and the variable holding cost. The limiting behavior of such a system was determined by "averaging over cycle", and formulas to compute the system parameters, i.e., the average cost per period and average quantity cleared, were constructed.

A variation of stochastic clearing systems, called "semi-stationary clearing processes",

was introduced by Serfozo and Stidham in [71]. Such systems still receive random input quantities over time and clear them intermittently, but these systems are strictly stationary over their random clearing epochs. Instead of the limiting probabilities, an asymptotic distribution was used to study such models, and the asymptotic distribution of the system state was proven to be uniform in some cases.

In [86], Whitt investigated a particular stochastic clearing problem, which is the utilization in capacity expansion, i.e., the utilization at any time divided by the capacity. Extending the results in [77], the stationary distribution of the clearing process is proven to be stochastically less than or equal to the uniform distribution in the sense of second-order stochastic dominance.

Inventory problems were touted as one of the main applications of stochastic clearing system. The results of stochastic clearing systems were generalized in $(s,\ S)$ inventory models by Stidham in [79]. Both the continuous-review and periodic-review versions of the inventory model were discussed. Necessary and sufficient conditions on the cost function and input process for optimality of the clearing parameters were identified.

Kim and Seila [52] showed that in the context of inventory systems, the optimal stochastic clearing policy outperforms the deterministic economic ordering quantity (EOQ) policy in certain scenarios. Their model has a more realistic cost structure which includes a clearing cost that is proportional to the number of items cleared, and they model the input process as a Brownian motion with positive drift. The optimal clearing level is shown to be the unique level at which the marginal cost of accepting another item is equal to the total average cost of operating the system.

The concepts of stochastic clearing systems were also applied to queueing systems. For example, Boxma and colleagues studied the clearing models for M/G/1 queues in [14], in which events called "disasters" occur at certain random times, causing an instantaneous removal of the entire residual workload from the system. In such queueing systems, clear-

ings can be at scheduled times, at random times, and at crossing of some pre-specified level. The stationary distribution of the workload process was derived.

In a similar line of research, Dudin and Karolik [30] considered BMAP/M/1 systems with potential exposure to disasters. The recovery process after a disaster is assumed to take place over a random time interval. Customers may or may not arrive during the recovery period. Using the concepts of stochastic clearing systems, the embedded and arbitrary time queue length distributions, as well as the average output rate and loss probability, were calculated.

In the recent literature, more complicated system parameters were put into the stochastic clearing framework. For example, in [51] Kella and colleagues considered a stochastic input-output system with additional total clearings at certain random times determined by its own evolution (and specified by a controller). Between two clearings, the stock level process is a superposition of a Brownian motion with drift and a compound Poisson process with positive jumps, reflected at zero. Cost functions for this system were introduced and determined explicitly under several (classical and new) clearing policies.

For other applications and modelling approaches for stochastic clearing systems, please refer to the following literature: [1, 2, 53, 74, 80, 87].

1.2.2 Shipment Consolidation Problems

Shipment consolidation is a logistics strategy whereby many small shipments are combined into a few larger loads. The economies of scale thus achieved help improve the utilization of logistics resources and reduce transportation costs. Such systems are one of the most natural examples of stochastic clearing systems.

Although the main purpose of shipment consolidation is to minimize overall costs, it should not be at the expense of unsatisfactory customer service. By associating appropriate monetary values to the delays of orders, achieving an optimal balance between cost

reductions and maintaining good service become the ultimate goal of that strategy. The shipment consolidation process can be categorized by *private carriage* and by *common carriage*. These respectively refer to the dispatch of a consolidated load in a company's own truck, or in the vehicle of an outside for-hire trucking company.

The shipment consolidation process is governed by a set of decision rules known as the "dispatch policies". These rules determine the appropriate size of the accumulated load, and/or the best time to release that load. Upon reaching the desired size or release time, those orders waiting are then sent, and the next cycle of the consolidation process begins anew.

Three classes of dispatch policies have been reported in the logistics literature. These are respectively the *quantity*, *time*, and *hybrid* policies. When a quantity policy is implemented, dispatch of a consolidated load is delayed until the total weight of those orders is at least Q; a time policy leads to a dispatch every T periods. A hybrid, or time-and-quantity-based, policy combines the effect of the previous two classes: there is still a desired shipment quantity Q, but if that weight is not attained by time T, those orders on hand are then dispatched. It is important to note that, in practice, there exist many other policies, e.g., those whose thresholds for dispatch may depend on order delays.

Over the years, several different operations research methods have been employed to study shipment consolidation problems. In [41], Higginson and Bookbinder used computer simulation to study a consolidation system with a Poisson arrival process and empirically-supported Gamma distribution of order weights. They examined the cost effectiveness of the three commonly used dispatch strategies. Their simulation results were based on a large range of the relevant parameters, long-run order arrival rates and maximum holding times. Using these results, they computed the cost per load, cost per hundredweight, and average order delay for each policy, and they made recommendations on how to choose the appropriate policy under different situations.

Higginson and Bookbinder continued their study of shipment consolidation by using the discrete-time Markovian decision process (MDP) approach for determining when to release consolidated loads in [42]. They considered two minimization criteria: cost per unit time, or cost per hundredweight per unit time. For private carriage, the optimal policy is of the control-limit type; for common carriage, it may not be. These potential differences in the structure of the optimal policy are true for either objective function. The possibly contrasting optimal policies are interpreted in light of the costs encountered by an industrial firm's private fleet compared to the freight charges of a public trucking company.

In their third paper on shipment consolidation [13], Bookbinder and Higginson employed probabilistic modelling to choose the maximum holding time and desired dispatch quantity. Practical decision rules for temporal consolidation for transportation by private carriage were obtained. They expressed the final results visually and conducted sensitivity analysis through a monograph (four linked graphs) relating decision variables, probability and demand parameters, and objective-function values.

Çetinkaya and Bookbinder [16] applied renewal theory to two consolidation strategies typically utilized in practice. For the case of a quantity policy, they obtained the optimal target weight before dispatch, while for a time policy, they calculated the optimal length of each consolidation cycle (maximum holding time for any order). These strategies were analyzed for private carriage and then for common carriage.

In [17], Çetinkaya and colleagues investigated the impact of shipment consolidation on the expected long-run average cost by simultaneously computing the optimal order quantity for inventory replenishment at the vendor and the optimal dispatch quantity for outbound shipments. They considered a case where demand follows a general stochastic bulk arrival process, and provided easy-to-compute approximations which enable efficient numerical solutions for the problem.

Mutlu and colleagues focused on the Time-and-Quantity (TQ)-based hybrid consoli-

dation policies in [58]. They developed an analytical model for computing the expected long-run average cost of a consolidation system implementing a TQ-based policy. Their analysis proved that the optimal TQ-based policy outperforms the optimal time-based policy, and the optimal quantity-based policy is superior to the other two (i.e., optimal time-based and TQ-based) policies in terms of cost. However, they also showed that the TQ-based policies improve on the quantity-based policies significantly in terms of timely delivery with only a slight increase in the cost.

Bookbinder and colleagues modelled the order arrival process by a discrete time batch Markov arrival process (BMAP) in [12]. The weight of an order was assumed to be discrete and may be correlated with the arrival time. A discrete time Markov chain for the accumulated weight of orders in the system was introduced and analyzed. The distributions of the accumulated weight at an arbitrary time, total accumulated weight in a consolidation cycle, and excess of weight per shipment were obtained. By introducing an absorbing Markov chain and a terminating Markovian arrival process, they found the distributions of the consolidation cycle length, the waiting time of an arbitrary order, and the number of orders that occur in a cycle. An efficient computational procedure was developed for evaluating dispatch policies.

Most recently in [15], Cai and colleagues proposed to use a tree structured Markov chain to record information about the consolidation process, specifically the quantities and waiting times of individual orders. The effect on shipment consolidation of varying the order-arrival process was demonstrated through numerical examples and proved mathematically under some conditions. A heuristic algorithm was developed to determine a favorable parameter of a special set of dispatch policies, and the algorithm was proved to yield the overall optimal policy under certain conditions. The results in this paper are presented in Chapter 6.

Other prominent publications on shipment consolidation include Chen et. al. [20], Dror and Martman [29], Gupta and Bagchi [33], Higgionson [40], Popkenl [65], and Tyan et. al.

[81].

1.2.3 Inventory Theories

Another example of a stochastic clearing system is the inventory control problem. In an inventory system, demands of random quantities are received at random time epochs and satisfied with the stock on hand, or backlogged until more inventories are received. Otherwise, the demand may be lost. Inventories are ordered in batches and received after some lead time. The cost structure of an inventory problem includes the inventory holding cost, backlog cost, and fixed and variable ordering costs.

Inventory control problems belong to the broader definition of stochastic clearing systems in two ways. First, if the inventory holding cost is too expensive for the system to keep any stock on hand, then all demands must be backlogged and satisfied later. This type of problem is called the *inventory backlog* problem, where the backlogged demands are the inputs in the stochastic clearing system, and instantaneous inventory replenishment is equivalent to the clearing of accumulated inputs.

More generally, we can treat any inventory replenishment as a clearing event, after which the inventory level is reset to a desirable level. Then, any demand received after the replenishment can be considered as an input to the system, and the total amount sold is the accumulated input quantity. At the next replenishment event, the clearing cycle begins anew. The inventory control problem is perhaps the most thoroughly studied stochastic clearing system. Therefore, we must review the literature on optimal inventory control.

One of the well-known inventory theories is the optimality of $(s,\ S)$ policies under certain conditions. Scarf introduced the concept of *K-convexity* in [70] and used it to prove the preceeding result. In that paper, he modelled the inventory problem using stochastic dynamic programming, and employed induction to prove the results. He also identified a set of conditions under which the optimality result holds.

In [45], Iglehart extended the optimality of (s, S) policies to the infinite horizon problem. He gave bounds for the sequences $\{s_n\}$ and $\{S_n\}$, discussed their limiting behavior, and established the existence of the limiting (s, S) policy. He also obtained similar results for the case of positive delivery lead time.

Veinott and Wagner developed a complete computational approach for finding optimal (s, S) inventory policies in [83]. The method was derived from renewal theory and stationary analysis. New upper and lower bounds on the optimal values of both s and S were established. The special case of linear holding and penalty costs was treated in detail.

In [82], Veinott constructed a different proof of the optimality of (s, S) policies under different conditions than those in [70]. The bounds on the optimal parameter values were established. Moreover, simple conditions were given which ensure that the optimal parameter values in a given period equal their lower bounds.

Zheng [90] presented a simpler alternative proof for the optimality of (s, S) policies in the infinite horizon cases which does not rely on the proofs for the finite horizon. With some novel arguments, he was also able to overcome the difficulties encountered in models with unbounded one-step expected costs.

Following these earlier works on the optimal inventory policies, many other theories have been developed for more complicated systems involving other supply chain decisions. For example, (s, S) policies are proven to be optimal in joint inventory-pricing control problems in Chen and Simchi-Levi [21], Chen et. al. [22], and Huh and Janakiraman [44]. Recent literature on inventory theory also studied systems with Markovian demand processes.

Iglehart and Karlin [46] were among the first to consider an inventory model with a stochastic demand process in which the distributions of demand in successive periods are not identical but in general are correlated. They used a discrete-time and continuous-state formulation with finite state space and linear ordering cost. Applying renewal theory and

stochastic dynamic programming, they proposed a policy which is determined by a set of critical numbers corresponding to the different demand distributions.

In [76], Song and Zipkin introduced a model where the demand rate varies with an underlying state-of-the-world variable. They derived some basic characteristics of the optimal policies and developed algorithms for computing them. They were also able to show that certain monotonicity patterns in the problem data can be reflected in the optimal policies.

Beyer and Sethi studied the inventory problem with Markovian demand with respect to the long-run average cost optimality criterion in [8]. They used the vanishing discount approach to establish a dynamic programming equation and the corresponding verification theorem. From that, they were able to prove the existence of an optimal state-dependent (s, S) policy.

In the context of expected total cost, Sethi and Cheng studied an inventory problem with Markovian demand in both the finite and infinite horizon cases [72]. They extended the optimality results to the state-dependent (s, S) policies, while also considering other realistic model features such as no-ordering periods and storage and service level constraints.

Chen and Song [19] extended the results further to multi-echelon inventory problems with Markov-modulated demand. They showed that for a linear ordering cost, the optimal policy is an echelon base-stock policy with state-dependent order-up-to levels, and built an algorithm to determine these levels. They extended their results to serial systems with a fixed ordering cost at the last stage.

In their book on Markovian demand inventory models [7], Beyer and colleagues provided a comprehensive summary of the results for this class of problems. The models they reviewed include finite and infinite horizon models under total cost, total discounted cost, and average cost objectives. In most of these problems, the optimal policies are proven to be state-dependent (s, S) policies.

Some of the well-known works on Markovian demand inventory models which Beyer

and colleagues summarized include: Bellman et. al. [3], Beyer and Sethi [9], Beyer et. al. [10, 11], Cheng and Sethi [23, 24], Porteus [66], and Zheng and Federgruen [91].

1.2.4 Other Related Problems and Theories

Stochastic clearing systems are related to several other well-studied operations research models. We shall look at their similarities and differences, and how some results from these other problems can be extended to the stochastic clearing systems.

The first type of problem is the dynamic lot-size problem. This problem is usually faced by a manufacturer who receives orders and plans to manufacture the products in batch production runs. There are fixed setup costs to begin a run, variable production costs for each item, and carrying costs for the work-in-progress and finished goods inventories. Similar to the clearing systems, the goal is to make the production batches into appropriate sizes so as to balance the production costs and inventory costs. However, in the lot-size problem, the orders are produced gradually over a period of time that is proportional to the size of the lot.

In their well-cited paper [84], Wagner and Whitin used a forward algorithm to solve the dynamic lot-size problem. They famously showed that disjoint planning horizons, which eliminate the necessity of having data for the full planning horizon, are possible. By studying the lot-size problem with non-zero chance of going out of control, Porteus [67] demonstrated that lower setup costs can benefit production systems by improving quality control. There is thus an incentive to produce smaller lots, and have a smaller fraction of defective units.

Fleischmann [32] studied the discrete lot-sizing and scheduling problem. Several products are to be scheduled on a single machine so as to meet known dynamic demand and to minimize the sum of inventory and setup costs. A branch-and-bound procedure was presented using Lagrangean relaxation for determining both lower bounds and feasible so-

lutions. The relaxed problems are solved by dynamic programming. Drexl and Kimms [28] summarized the work in the field of lot sizing and scheduling, and identified two research directions: continuous time models and multi-level lot sizing and scheduling. Other works on the dynamic lot-size problem include [34], [49], and [89].

In mathematics, the theory of optimal stopping is concerned with the problem of choosing a time to take a particular action, in order to maximize an expected reward or minimize an expected cost. Optimal stopping problems can be found in the areas of statistics, economics, and mathematical finance (related to the pricing of American options). The optimal stopping problems can often be written in the form of a Bellman equation, and are therefore often solved using dynamic programming. In a sense, the optimal stopping problem is very similar to the stochastic clearing system, in which the stopping time is the time at which the system is cleared.

There are generally two approaches for solving optimal stopping problems. When the underlying process is described by its unconditional finite-dimensional distributions, martingale theory is used. In the discrete time case, if the planning horizon T is finite, the problem can also be easily solved by dynamic programming. When the underlying process is determined by a family of (conditional) transition functions leading to a Markovian family of transition probabilities, theories of Markov processes can be utilized and the solution is usually obtained by solving the associated free-boundary problems (Stefan problems).

Some of the important works on optimal stopping problems include Chow et. al. [25], Jacka [48], Karatzas [50], and Shiryaev [75]. These publications are primarily in the context of mathematical finance, and use advanced tools and methodologies such as Brownian motion, Martingales, stochastic calculus, and diffusion processes. These methods are out of the scope of our current research, but they can be explored and potentially applied to the stochastic clearing problem in the future.

1.2.5 Potential Research Problems

As we can see from the previous literature review, stochastic clearing systems were first studied as a generalized model when the modelling framework and cost structure were set up in the 1970's and 1980's. Since then, the research has been focused on separate application areas of stochastic clearing systems, such as shipment consolidation problems and inventory control problems. Due to the varying nature of these application areas, many different modifications of the basic model of stochastic clearing systems have been constructed and analyzed. However, all of these models in the literature have assumed that the holding cost of the accumulated inputs is charged at a *constant* rate, proportional to the total quantity accumulated over time.

This common assumption about the holding cost structure is sufficient in areas such as inventory control, where the time spent in the system by individual inputs has no effect on the rate of the holding cost. Unfortunately, in other areas such as shipment consolidation and shuttle bus dispatch, the longer an input stays in the system, the more expensive the holding cost rate tends to become. For instance, this kind of situation exists in all four motivating examples of stochastic clearing systems we described at the beginning of this chapter.

If we look at that situation from the modelling perspective, we can easily see how the current literature on stochastic clearing systems is ill equipped to reflect this delicate situation. Let us consider two instances faced by the same clearing system. In the first case, four units of input were received three periods ago, while another unit just arrived only in the last period. In the second case, one unit was received three periods ago, and four other units have just arrived in the last period. It is not hard to see that the first situation is more urgent than the second as a higher proportion of the input has been held back for a longer time. However, the traditional holding cost structure would not be able to distinguish the two cases, because in any successive period before clearing, the total

accumulated quantity is five units in either case. Thus, we identify this deficiency of the holding cost structure in the current literature as one of the main problems we want to solve in our research here.

The other main research goal we would like to address is to incorporate a Markovian input process into the general model of stochastic clearing systems. That this is important because a Markovian input process can better model input processes which are affected by some underlying states of the world or exhibit cycles and seasonality effects. Since Markovian processes are also known to be able to approximate any stochastic input processes, this would lend more generality to our model.

1.3 Methodology

In this section, we shall review some methodologies which will be used in our research.

1.3.1 Markov Decision Processes and Stochastic Dynamic Programming

Markov decision processes, also referred to as stochastic dynamic programs or stochastic control problems, are models for sequential decision making when outcomes are uncertain. In the framework of stochastic clearing systems, the clearing decisions, i.e., clear the system vs. continue to accumulate, must be made in each period.

The Markov decision process model consists of five major components: decision epochs, states, actions, rewards/costs, and transition probabilities. At each decision epoch, an action is decided based on the state of the system and a reward/cost is generated. Then the system evolves according to the action chosen and the transition probabilities associated with the current state and action. The process continues over a finite or infinite planning

horizon. Policies or strategies are prescriptions of which action to choose under any eventuality at every future decision epoch. The goal of the decision maker is to seek policies which are *optimal* in some sense.

The general approaches in analyzing Markov decision processes include: (i) providing the optimality conditions for easily implementable optimal policies, (ii) determining if there are any special characteristics about the optimal policies, (iii) developing and enhancing algorithms for computing the optimal policy parameters, and (iv) establishing convergence of these algorithms.

Standard Markov decision processes are usually modelled in discrete time and discrete state space. The discrete settings allow us to keep track of the system state and ensure that the action sets are of manageable size. The special class of continuous-time and discrete-state models is referred to as a semi-Markov decision process, for which the system state evolves as a Markov chain only when certain events occur.

The planning horizon, i.e., consecutive decision epochs, can be either finite or infinite, and depending on the length of that horizon, different objective functions can be used. For example, in finite horizon problems, the decision maker is often trying to optimize the expected total reward/cost. On the contrary, in infinite horizon problems, reward/cost can be discounted, and in the long run, the expected average reward/cost may be a more appropriate measure of the asymptotic behavior of the system.

In most studies of Markov decision processes, induction proofs are used to verify results in a finite horizon. Then, these results are extended to the infinite horizon by analyzing the limits of the finite horizon functions. For long-run average analysis, the stationary distribution of the Markov chain which governs the system state transitions can be used to compute long-run average measures.

There are two main methods to compute the optimal policies: value iteration and policy iteration. The value iteration algorithm was first introduced by Bellman [4] and is

also known as *backward induction*. This type of algorithm relies on solving for the *value function* of the process, i.e., a solution to the optimality equations, and then constructing the optimal policy based on the value function. The policy iteration algorithm begins with an arbitrary policy, then derives the optimal policy through successive evaluations and improvements.

Note that the choice of action depends on the current state, but if that state is not observable or is only partially observable, then the system becomes a partially observable Markov decision process (POMDP), which is a generalization of a Markov decision process. A POMDP models an agent decision process in which it is assumed that the system dynamics are determined by an MDP, but the agent cannot directly observe the underlying state. Instead, he/she must maintain a probability distribution over the set of possible states, based on a set of observations and observation probabilities, and the underlying MDP.

The POMDP framework is general enough to model a variety of real-world sequential decision processes. Applications include robot navigation problems, machine maintenance, and planning under uncertainty in general. This framework originated in the operations research community, and was later taken over by the artificial intelligence and automated planning communities.

An exact solution to a POMDP yields the optimal action for each possible belief over the world states. The optimal action maximizes (or minimizes) the expected reward (or cost) of the agent over a possibly infinite horizon. The sequence of optimal actions is known as the optimal policy of the agent for interacting with its environment.

In our research, we have consulted the following books on the topic of Markov decision processes Bellman [4], Bertsekas [5], Bertsekas and Shreve [6], Derman [27], Hillier and Liebman [43], and Puterman [68]. These books contain a vast array of literature and summarize the key results and algorithms obtained over the years.

1.3.2 Markov Chain and Matrix Analytic Methods

In this section, we will review some modelling tools and solution techniques involving the use of Markov chains, including the Markovian arrival processes and matrix-analytic methods. These tools and techniques will help us model stochastic clearing systems with greater generality, and solve the problem more accurately and efficiently.

Although the input process in a clearing system has traditionally been assumed to be a Poisson process in the existing literature, we would like to model it with a more general stochastic arrival process. A Markovian arrival process (MAP) seems to be a good choice. The MAP was initially introduced by Neuts [59] as a versatile class of point processes on the real line. A MAP is usually denoted by a pair of matrices (D_0, D_1), where elements of D_0 can be interpreted as the (event) rates without an arrival, and elements of D_1 as the rates with an arrival. One of the advantages of using MAPs for stochastic modelling and analytical research is the fact that the relevant probability distributions, moment and correlation formulas are given in forms which are computationally tractable. More importantly, MAP is so versatile that it can closely approximate any stochastic counting process.

According to He [37], a MAP is a generalization of the Poisson process. It not only keeps many useful properties of the Poisson process, it partially preserves the memoryless property by conditioning on the phase of an underlying Markov chain. MAP also includes several other well-known point processes, such as renewal processes of phase type, Markov-modulated Poisson processes, and certain semi-Markov point processes. Such generality will allow us to model different arrival characteristics.

The batch Markovian arrival process (BMAP) and the marked Markovian arrival process (MMAP) are direct generalizations of MAP. They are usually defined by splitting the arrival rates in the D_1 matrix of a MAP, and assigning different meanings to these rates. In the context of stochastic clearing systems, we can assign these arrival rates to different

weight categories or even distinct combinations of inputs. BMAPs and MMAPs are crucial components in our models that follow.

Through simple intuition, we can claim that the future state of a clearing process only depends on its current state, the clearing decision, and future inputs. This is a memoryless property, otherwise known as the Markov property. Thus, we can model the stochastic clearing process by a Markov chain.

A Markov chain is defined for a discrete (or continuous) set of times. The state of the Markov chain evolves over time and the future state depends only on the current state, but not on the past. If a Markov chain is ergodic (i.e. aperiodic and positive recurrent), then its limiting probabilities (a.k.a. stationary distribution) are unique and well defined.

Markov chains have many applications as statistical models of real-world processes. Past works on shipment consolidation, such as [42] and [12], have used Markov chains to model the clearing system, especially when the goal is to assess long-run performance. In our current research, Markov chains will again be used to model the clearing process, albeit the states of our Markov chain will be more elaborate and contain more information.

In stochastic modelling, we often encounter Markov chains of large dimension, for which we have to solve for their stationary distributions. Consider a discrete time Markov chain with transition probability matrix P. Even if P is sparse, but has very large dimension, it can be challenging to solve for its stationary distribution $\boldsymbol{\pi}$. However, if P has some special block structures, we can use an algorithmic approach to compute $\boldsymbol{\pi}$. Matrix Analytic Methods (MAM) consititute a set of modelling/computational tools that "give one the ability to construct and analyze, in a unified way and in an algorithmically tractable manner, a wide class of stochastic models" [54].

In our research, we will attempt to utilize such methods to build efficient models and algorithms. Several methods may be of particular interest to us. They include matrix geometric solution, algorithms for the rate matrix R, fundamental periods, and Markov

chains with tree structures.

For more details on MAP, BMAP, MMAP and MAM, we will refer to Neuts [59, 60, 61, 62], Lucantoni [56], Yeung and Sengupta [88], He and Neuts [38], Latouche and Ramaswami [54], He [35, 36, 37], and Latouche et. al. [55].

1.4 Thesis Layout

The remainder of the thesis is arranged as follows. In Chapter 2, we describe the general model that is going to be used throughout the thesis. Modelling assumptions and cost structures are given and two Markov chains are introduced to keep track of the process.

We begin our analysis of the model in the case of expected total cost over a finite horizon in Chapter 3. We show some characteristics of the optimal policies, introduce a threshold-based policy which can be optimal under certain conditions, and finally, build an optimization algorithm.

We then consider an infinite horizon model, and show that the results from Chapter 3 can be extended in Chapter 4. Based on the special structure of our system state space, we modify the value iteration algorithm and policy iteration algorithm for standard MDP to help find the optimal clearing policies.

In Chapter 5, we change the model objective to expected average cost over an infinite horizon and use the MDP approach to construct the optimality equations and build three optimization algorithms.

For the same model studied in Chapter 5, we use the MAM approach to analyze it in Chapter 6. Taking advantage of the special structure of the system state space, we construct an efficient algorithm to evaluate a given policy in terms of expected average cost and other performance measures. We also propose a search algorithm to compute the optimal threshold-based policies, and an iterative policy construction algorithm for

building the optimal policies.

Lastly, a summary of the main results in this thesis is given in Chapter 7 and some future research topics on stochastic clearing systems are described as well.

Chapter 2

Model Formulation

In this chapter, we introduce the general model that is used throughout the thesis. The definitions and notations of each model component are described in detail, and some examples are given to demonstrate the exact model parameters under different circumstances.

2.1 Problem Definition and Notation

To model any stochastic clearing system, we need to (i) define the time interval and frequency over which to analyze the clearing process, (ii) observe and model the process in which random input quantities are received by the system, (iii) choose the amount of detail we want to measure and record about the state of the system, and (iv) translate the desired clearing criteria into a well-defined clearing policy. We examine each of the above components in the following sections.

2.1.1 Decision Epochs, Planning Horizon, and Periods

For any stochastic clearing system, the clearing decisions are made at points of time referred to as the *decision epochs*. The set of decision epochs to be considered is referred to as the

planning horizon, denoted by T. Since it is a representation of time, the planning horizon is a subset of the nonnegative real line, i.e., $T \in \mathbb{R}_{\geq 0}$. We can classify T as either a discrete set or a continuum, but in the scope of our research, we assume T to be discrete. This is not very restrictive in practical examples of stochastic clearing systems, such as those described in Section 1.1.

In discrete time problems, time is divided into *periods*. The appropriate length of a period is chosen to suit specific problems. In all our models, we assume that clearing decisions are made at the beginning of each period, so a decision epoch corresponds to the beginning of a period. Figure 2.1 shows a sample time line of the clearing process.

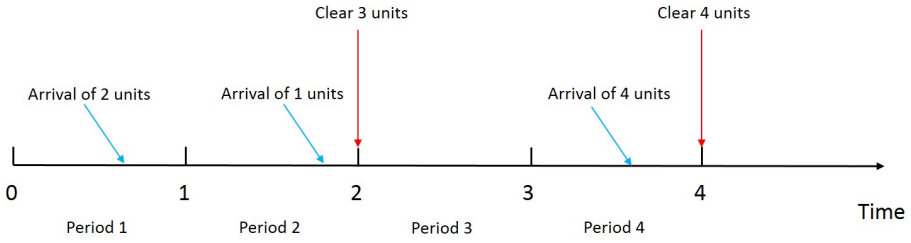

Figure 2.1: A sample time line of the stochastic clearing process

The set of decision epochs is either finite or infinite, we write $T \equiv \{1, 2, \ldots, N\}$, for some integer $N \leq \infty$, to include both cases. Individual elements of T will be denoted by t and referred to as "time/period" t. We call the problem a *finite horizon* problem when T is finite, and an *infinite horizon* problem otherwise. Our research will cover both the finite and infinite horizon problems.

The convention for finite horizon problems requires an evaluation of the final system state at the end of the planning horizon. Thus, we include an extra decision epoch $N + 1$, at which point no decisions are made. In the context of stochastic clearing systems, all accumulated inputs must be cleared at time $N + 1$. We frequently refer to this as an N-period problem.

2.1.2 Input Processes

In a stochastic clearing system, *inputs* of random quantities are received by the system and cleared at a later time. For discrete time models, we tally the total quantity received in each period, and if such quantity is zero, we recognize that no input is received in the period. Before an input is cleared from the system, we refer to it as an *outstanding input* in the system.

In our research, we study input processes that are governed by some *underlying states of the world*. Let $I = \{1, 2, \ldots, M\}$ denote the collection of all such states, for which we assume that $M < \infty$ to keep our models mathematically tractable. Let the random variable i_t represent the underlying state at the beginning of period t. We assume that the corresponding stochastic process $\{i_t, t = 1, 2, \ldots\}$ is an ergodic Markov chain with $M \times M$ transition matrix D.

Let q_t be the random quantity of all inputs received during period t, and obviously, q_t is nonnegative, i.e., $q_t \in \mathbb{R}_{\geq 0}$. Suppose that q_t only depends on the underlying state i_t. We can easily verify that the *stochastic input process*, $\{(q_t, i_t), t = 1, 2, \ldots\}$, is a Markovian input process. The *conditional input quantity*, denoted by $[q_t | i_t = i]$, is a random variable with conditional probability distribution

$$F_i(q) = Pr\{q_t \leq q | i_t = i\}. \tag{2.1}$$

We further assume that the input quantity received in each period is nonnegative and bounded, i.e., $0 \leq q_t \leq Q < \infty, \forall t$. This is not a restrictive assumption from the practical perspective since for most stochastic clearing systems, the input quantities are finite, and often even relatively small.

The value of q_t can be discrete or continuous depending on the specific problem. However, for some of the algorithms developed in the later chapters, it helps to have discrete

input quantities to keep our models mathematically tractable, and these models are referred to as *discrete time and discrete quantity models*. It is worth noting that in many stochastic clearing systems, continuous input quantities are often rounded to discrete units by the decision makers. For example, in some shipment consolidation examples, the input quantities are measured as shipment weights and rounded to the nearest pound, kilogram, or ton.

For discrete time and discrete quantity models, the above Markovian input process can be modelled as a discrete time *batch Markovian arrival process* ($BMAP$) with matrix representation $(D_0, D_1, \ldots, D_Q)$, where $D_0, D_1, \ldots, D_Q$ are $M \times M$ nonnegative matrices.

Entry (i, j) in matrix D_q, denoted by $[D_q]_{i,j}$, for $q = 0, 1, \ldots, Q$, can be interpreted as the probability that q units of input have been received in a period, and the underlying process goes from state i at the beginning of the current period to state j at the beginning of the next period, i.e.,

$$Pr\{q_t = q, i_{t+1} = j | i_t = i\} = [D_q]_{i,j}. \tag{2.2}$$

Consequently,

$$Pr\{q_t = q | i_t = i\} = \sum_{j \in I} [D_q]_{i,j} = f_i(q), \tag{2.3}$$

where $f_i(q)$ is the probability mass function for $[q_t | i_t = i]$. On the other hand, we have

$$Pr\{q_t = q | i_t = i, i_{t+1} = j\} = \frac{[D_q]_{i,j}}{\sum_{q=0}^{Q} [D_q]_{i,j}}, \tag{2.4}$$

and finally

$$\sum_{q=0}^{Q} [D_q]_{i,j} = [D]_{i,j}, \tag{2.5}$$

where $[D]_{i,j}$ is the transition probability from state i to state j in the underlying Markov chain $D = D_0 + D_1 + \ldots + D_Q$.

Let $\boldsymbol{\theta}_{\mathrm{a}}$ be the stationary distribution of the underlying Markov chain. Then $\boldsymbol{\theta}_{\mathrm{a}}$ is the unique solution to the linear system $\boldsymbol{\theta}_{\mathrm{a}} D = \boldsymbol{\theta}_{\mathrm{a}}$ and $\boldsymbol{\theta}_{\mathrm{a}} \boldsymbol{e} = 1$, where $\boldsymbol{e}$ is a column vector of 1's with the appropriate dimension. Define the *quantity-receiving rate*, i.e., the average quantity received in each period in the steady state, as

$$\lambda_{\mathrm{q,a}} = \boldsymbol{\theta}_{\mathrm{a}} \left(\sum_{q=1}^{Q} q D_q \right) \boldsymbol{e}, \tag{2.6}$$

and let the *input-receiving rate*, i.e., the average rate of receiving any positive input quantities in the steady state, be

$$\lambda_{\mathrm{o,a}} = \boldsymbol{\theta}_{\mathrm{a}} \left(\sum_{q=1}^{Q} D_q \right) \boldsymbol{e}. \tag{2.7}$$

The following Markovian input processes can be modelled as special cases of *BMAP*.

(i) Compound Renewal Process:

$$(D_0, D_1, \ldots, D_Q) = (d_0, d_1, \ldots, d_Q), \tag{2.8}$$

where $0 \leq d_q \leq 1, \forall q = 1, \ldots, Q$ and $\sum_{q=0}^{Q} d_q = 1$. This process has a single underlying state in which the input quantities for each period are independent and identically distributed (i.i.d.) random variables, and the process is a discrete analogue of the compound Poisson process.

(ii) Markov Modulated Input Process:

$$D_q = \begin{pmatrix} f_1(q) & & \\ & \ddots & \\ & & f_M(q) \end{pmatrix} D, \forall q = 0, 1, \ldots, Q, \tag{2.9}$$

where $f_i(q) = Pr\{q_t = q | i_t = i\} = \sum_{j \in I} [D_q]_{i,j}$, for $i \in I$. This is equivalent to the typical Markovian input process introduced earlier.

(iii) Compound Markovian Arrival Process:

$$(D_0, D_1, \ldots, D_Q) = (D_0, \tilde{f}(1)D_1', \ldots, \tilde{f}(Q)D_1'), \tag{2.10}$$

where $\tilde{f}(q) = Pr\{q_t = q | q_t > 0\}, \forall q = 1, \ldots, Q$ and $\sum_{q=1}^{Q} D_q = D_1'$. In this process, the underlying states affect the probabilities that some positive input is received in a period, but not the actual quantity of that input.

Example 2.1.1. *Numerical examples of the above special cases of the BMAP input processes are presented as follows*

(i) Compound Renewal Process:

$$(D_0, D_1, D_2, D_3) = (0.25, 0.25, 0.25, 0.25)$$

(ii) Markov Modulated Input Process:

$$D_0 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.6 \end{pmatrix} \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix} = \begin{pmatrix} 0.09 & 0.01 \\ 0.06 & 0.54 \end{pmatrix}$$

$$D_1 = \begin{pmatrix} 0.3 & 0 \\ 0 & 0.3 \end{pmatrix} \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix} = \begin{pmatrix} 0.27 & 0.03 \\ 0.03 & 0.27 \end{pmatrix}$$

$$D_2 = \begin{pmatrix} 0.6 & 0 \\ 0 & 0.1 \end{pmatrix} \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix} = \begin{pmatrix} 0.54 & 0.06 \\ 0.01 & 0.09 \end{pmatrix}$$

(iii) Compound Markovian Arrival Process:

$$D_0 = \begin{pmatrix} 0.99 & 0 \\ 0.01 & 0 \end{pmatrix}$$

$$D_1 = 0.2 \begin{pmatrix} 0 & 0.01 \\ 0 & 0.99 \end{pmatrix} = \begin{pmatrix} 0 & 0.002 \\ 0 & 0.198 \end{pmatrix}$$

$$D_2 = 0.3 \begin{pmatrix} 0 & 0.01 \\ 0 & 0.99 \end{pmatrix} = \begin{pmatrix} 0 & 0.003 \\ 0 & 0.297 \end{pmatrix}$$

$$D_3 = 0.5 \begin{pmatrix} 0 & 0.01 \\ 0 & 0.99 \end{pmatrix} = \begin{pmatrix} 0 & 0.005 \\ 0 & 0.495 \end{pmatrix}$$

2.1.3 System Contents

Any model of stochastic clearing system must keep track of the total quantity of all outstanding inputs. Our model extends the existing models by simultaneously recording the *ages* of individual inputs, i.e., the time elapsed since each input was received. This piece of additional information is useful if the inputs or clearing decisions are time sensitive. For instance, in the shuttle bus clearing example described in Section 1.1, passengers may leave via other modes of transportation if they have been waiting for too long, and the lost revenue can be thought of as a penalty cost.

Therefore, the system state of our model includes both the quantities and the ages of individual outstanding inputs, and this information can be recorded in a *sequence* (or *string*) of numbers. Let $\mathbb{R}^l_{\geq 0}$ denote the set of all nonnegative number sequences of length l and let $\mathbb{R}^0_{\geq 0} = \emptyset$. Any such sequence $x \in \mathbb{R}^l_{\geq 0}$ can be written as $x = \big[x_{[l]}, x_{[l-1]}, \ldots, x_{[1]}\big]$.

Now let $x_t \in \bigcup_{l=0}^{\infty} \mathbb{R}^l_{\geq 0}$ record the sequence of all outstanding inputs at the beginning

of period t. Since this state of the system is captured before any clearing decision is made, we shall refer to it as the *pre-clearing system content* in period t. Suppose that $x_t = \big[x_{[l]}, x_{[l-1]}, \ldots, x_{[1]}\big]$. Then, for each $j = 1, 2, \ldots, l$, we have $0 \leq x_{[j]} \leq Q$, and $x_{[j]}$ is the j-th number starting from the right (or back) of the sequence representing the outstanding input received in period $(t - j)$; the subscript j denotes the age of this input in period t. Note that $x_t = \emptyset$ (or $x_t = [0, \ldots, 0]$) indicates that the system is currently empty.

Analogous to x_t, we can define the *post-clearing system content* as y_t, where $y_t \in \bigcup_{l=0}^{\infty} \mathbb{R}^l_{\geq 0}$, i.e., the sequence of outstanding inputs remaining in the system immediately after the clearing decision is carried out. The difference between x_t and y_t is the outstanding input quantity that is cleared in period t, which we shall denote as w_t.

The system content variables x_t and y_t are both affected by the underlying state of the system, the input process, and the clearing decision. We can use either variable to keep track of the system state depending on the specific needs, and we shall distinguish them when necessary.

Under the context of our model, the system remains empty until the first positive input is received. Therefore, it is important to note that for any sequence, consecutive entries of zeros not "preceded" by a positive number, i.e., zeros to the left (or at the front) of the sequence, can be truncated without affecting the meaning of the sequence. In other words, any two sequences such as $\big[0, \ldots, 0, x_{[l]}, \ldots, x_{[1]}\big]$ and $\big[x_{[l]}, \ldots, x_{[1]}\big]$ contain essentially the same information and can be used interchangeably in our models. With that said, the set of potential system contents is defined by

$$\Phi = \{\emptyset\} \cup \bigcup_{l=1}^{\infty} \{\big[x_{[l]}, \ldots, x_{[1]}\big] : x_l > 0\}. \tag{2.11}$$

Together with the underlying state, we define the set of potential system states as

$$\Omega = \Phi \times I. \tag{2.12}$$

Note: For discrete time and discrete quantity models, Φ can be arranged into a tree structure. An element in the tree is called a *node*, and the root node in the tree represents the empty system. For a particular state x on the tree, all of its *successors*, i.e., nodes found on any downward path from x, can be reached after receiving a sequence of inputs without clearing the system. Therefore, each downward path of the tree corresponds to a sample path of the input accumulation process. Figure 2.2 is an example of the tree of Φ with the maximum input quantity $Q = 2$. The dashed lines represent the existing subtrees not shown in the figure.

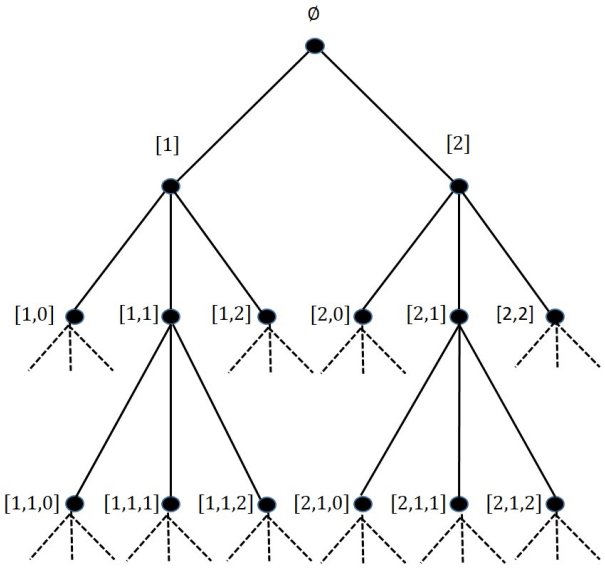

Figure 2.2: A sample tree of Φ with $Q = 2$

Definition 2.1.1. *For any pair of sequences* $x = \left[x_{[l]}, x_{[l-1]}, \ldots, x_{[1]}\right]$ *and* $y = \left[y_{[k]}, y_{[k-1]}, \ldots, y_{[1]}\right]$ *in* Φ*, we define the following operators for elements of* Φ*:*

(i) sequence concatenation:

$$\begin{aligned} x \oplus y &= \left[x_{[l]}, x_{[l-1]}, \ldots, x_{[1]}, y_{[k]}, y_{[k-1]}, \ldots, y_{[1]}\right] \\ &= \left[z_{[l+k]}, z_{[l+k-1]} \ldots, z_{[1+k]}, z_{[k]}, z_{[k-1]} \ldots, z_{[1]}\right] = z \end{aligned}$$

(ii) sub-sequences:

$$x_{\mathrm{L}}(n) = \left[x_{[l]}, x_{[l-1]}, \ldots, x_{[l-n+1]}\right] \text{ and } x_{\mathrm{R}}(n) = \left[x_{[n]}, x_{[n-1]} \ldots, x_{[1]}\right], \text{if } 1 \leq n \leq l$$

(iii) sequence scalar multiplication:

$$ax = \begin{cases} \emptyset, & \text{if } a = 0 \\ \left[ax_{[l]}, \ldots, ax_{[1]}\right], & \text{if } a \in \mathbb{R} \setminus \{0\} \end{cases}$$

(iv) sequence addition:

$$x + y = \begin{cases} \left[x_{[l]}, \ldots, x_{[k+1]}, x_{[k]} + y_{[k]}, \ldots, x_{[1]} + y_{[1]}\right], & \text{if } l \geq k \\ \left[y_{[k]}, \ldots, y_{[l+1]}, x_{[l]} + y_{[l]}, \ldots, x_{[1]} + y_{[1]}\right], & \text{if } l < k \end{cases}$$

(v) sequence sum: $|x| = \sum_{j=1}^{l} x_{[j]}$

(vi) sequence length: $\mathcal{L}(x) = l$

Note: $\oplus$ is non-commutative, i.e., $x \oplus y \neq y \oplus x$; $\emptyset \oplus 0 = 0 \oplus \emptyset = \emptyset$ are two special cases of the concatenation operation; $x_{\mathrm{L}}(n)$ and $x_{\mathrm{R}}(n)$ are the left and right sub-sequences of x of length n, respectively; and the sequence sum is in fact a norm of Φ.

According to the above definitions, we can easily verify that each $\mathbb{R}^l_{\geq 0}$, for $l = 0, 1, \ldots$, as well as Φ, are all vector spaces with the sequences of zeros or $\emptyset$ being the zero elements, respectively. Moreover, Φ is a *metric space* if we use the sequence sum as the metric.

For the purpose of our research, we need to define a *partial order* on Φ, which we shall call the *sequence order*.

Definition 2.1.2. *For any pair of number sequences $x, y \in \mathbb{R}^l$, $x \leq y$ if and only if $x_{[j]} \leq y_{[j]}$, for all $j = 1, 2, \ldots, l$.*

The sequence order is essentially a number-by-number comparison of two sequences of the same length. However, since two sequences may vary in length, we need to "attach" zeros to the left of the shorter one to make the two sequences equal in length if necessary. Thus, we can relax our definition to allow comparisons of any pair of sequences in Φ. In addition, $x \not\leq y$ means that for some $x \in \mathbb{R}^l$ and $y \in \mathbb{R}^k$, we have at least one j, $1 \leq j \leq \max\{l, k\}$, such that $x_{[j]} > y_{[j]}$.

2.1.4 Action Set, Decision Rules, and Clearing Policies

At each decision epoch, we must decide whether to continue accumulating inputs, clear some outstanding inputs, or clear everything in the system. Let a_t denote the action taken and A_t be the set of available actions at the beginning of period t. Since we can clear any amount of outstanding inputs, the content of A_t depends on the pre-clearing system content x_t.

If not all outstanding inputs are cleared at once, we call the action a "partial clearing". When partial clearing is not allowed, and it is decided to clear, then the system must be reset to the empty state. This kind of action is called "bang-bang control" in the context of optimal control theory, in which case the optimal control switches from one extreme to the other (i.e., clear nothing vs. clear everything). Therefore, in the case of bang-bang control, where no partial clearing is allowed, we can let $A_t = \{0, 1\}$ for all $t \in T$, such that $a_t = 0$ means to continue to accumulate in period t and $a_t = 1$ means to clear the system immediately.

According to [68], "a decision rule prescribes a procedure for action selection in each state at a specific decision epoch". Decision rules may vary depending on how they incorporate past information and how they select actions. We say a decision rule is *deterministic* if it chooses an action with certainty, and *randomized* otherwise. We can also refer to a decision rule as *Markovian* if it only depends on the current system state, and *history dependent* if it depends on the past history of the system states and actions.

Following the notation introduced in [68], we can classify the decision rules as history dependent and deterministic (HD) and Markovian and deterministic (MD). In the context of stochastic clearing systems, we refer to the decision rule in period t as the *clearing rule*, model it by function r_t, and denote the set of all such clearing rules by R_t. Since the input process is Markovian, we can prove that Markovian clearing rules are sufficient for the purpose of our model. Hence, we focus attention on MD rules unless otherwise specified.

Each clearing rule specifies a set of "undesirable" system states in which the clearing action must take place in period t. Some typical rules include: (i) clear system if the total cumulative quantity reaches a certain level; (ii) clear system if the age of any outstanding order exceeds a particular limit; (iii) a combination of the preceding; and (iv) clear the system when it is in certain underlying states. In our models, the MD clearing rules are modelled as functions $r_t : \Omega \to A_t$. For bang-bang control, $r_t(x, i) = 0$ means to continue to accumulate in period t and $r_t(x, i) = 1$ means to clear the system immediately.

Example 2.1.2. *Here are some examples of those typical clearing rules*

(i) Quantity threshold rule:

$$r_t(x_t, i_t) = \begin{cases} 0, & \text{if } |x_t| < 5 \\ 1, & \text{otherwise} \end{cases}$$

(ii) Age threshold rule:

$$r_t(x_t, i_t) = \begin{cases} 0, & \text{if } \mathcal{L}(x_t) < 4 \\ 1, & \text{otherwise} \end{cases}$$

(iii) Underlying state rule:

$$r_t(x_t, i_t) = \begin{cases} 0, & \text{if } |x_t| < 5 \text{ and } \mathcal{L}(x_t) < 4 \\ 1, & \text{otherwise} \end{cases}$$

(iv) Hybrid rule:

$$r_t(x_t, i_t) = \begin{cases} 0, & \text{if } i_t \notin I^* \subset I \\ 1, & \text{otherwise} \end{cases}$$

According our definition of the clearing process, for bang-bang control, we have

$$y_t = x_t \cdot (1 - r_t(x_t, i_t)) = x_t \cdot (1 - a_t), \tag{2.13}$$

and

$$w_t = x_t - y_t = x_t \cdot r_t(x_t, i_t) = x_t \cdot a_t. \tag{2.14}$$

Consequently, at the next decision epoch $t + 1$

$$x_{t+1} = y_t \oplus q_t = \begin{cases} x_t \oplus q_t, & \text{if } a_t = 0 \\ q_t, & \text{otherwise} \end{cases}. \tag{2.15}$$

A *clearing policy* specifies the clearing rules to be used at all decision epochs. It provides the decision maker with a prescription for action selection in any system state or with respect to system history. A policy π is a series of clearing rules, i.e., $\pi = (r_1, r_2, \ldots, r_N)$, where $r_t \in R_t$, for all $t = 1, 2, \ldots, N$. We let Π denote the set of all clearing policies such

that

$$\Pi = R_1 \times R_2 \times \ldots \times R_N. \tag{2.16}$$

Note: We call a policy *stationary* if $r_t = r$, for all $t = 1, 2, \ldots, N$. A stationary policy over an infinite horizon has the form $\pi = (r, r, \ldots)$. In many infinite horizon problems, stationary policies are known to be effective and necessary to keep the policy tractable [68]. Therefore, we consider only stationary policies for infinite horizon problems. All definitions and results obtained in this section can be extended to stationary policies for an infinite horizon problem by simply dropping the subscript t.

One of the main objectives of our research is to find the optimal clearing policy for a given optimality criteria, and we denote such policy as π^*. Note that for a set of optimality criteria, the optimal clearing policy may not be unique because in certain state, it may be optimal to either continue to accumulate or clear all inputs.

2.2 Two Markov Chains

In Section 2.1.3, we introduced two types of system content variables x_t and y_t. Together with the underlying state of the world, the system state at the beginning of period t before clearing is given by the two-tuple of (x_t, i_t), and we call $\{(x_t, i_t), t = 1, 2, \ldots\}$ the *pre-clearing system state process*. Similarly, we call $\{(y_t, i_t), t = 1, 2, \ldots\}$ the *post-clearing system state process*. In this section, we show that under bang-bang control clearing policies, both processes can be modelled as GI/M/1-type Markov chains with tree structure. In the rest of this section, all clearing policies are assumed to be bang-bang control policies.

2.2.1 System State Spaces

By definition, the pre-clearing system state (x_t, i_t) is affected by the clearing rule r_{t-1} in the previous period. On the other hand, the post-clearing system state (y_t, i_t) is determined

by r_t. We can define the system state space for (x_t, i_t) and (y_t, i_t) in period t as follows.

Definition 2.2.1. *Let*

$$\Lambda_{(r_{t-1})} = \bigcup_{q \in \{0,\ldots,Q\}} \{(x \oplus q, i) \in \Omega : r_{t-1}(x, i) = 0\}, \tag{2.17}$$

be the system state space of (x_t, i_t)*, for* $t = 2, 3, \ldots$*. Similarly, let*

$$\Omega_{(r_t)} = \{(x, i) \in \Omega : r_t(x, i) = 0\}, \tag{2.18}$$

be the system state space of (y_t, i_t)*, for* $t = 1, 2, \ldots$*.*

Note: If the clearing rules vary over time, we must remember that $\Lambda_{(r_t)}$ corresponds to x_{t+1} and $\Omega_{(r_t)}$ corresponds to y_t. We do not need to define the system state space for (x_1, i_1) because that is determined by the model parameters. If the clearing rules are stationary, then by dropping the subscript t, only $\Lambda_{(r)}$ and $\Omega_{(r)}$ are needed to represent the system state spaces for $\{(x_t, i_t), t = 1, 2, \ldots\}$ and $\{(y_t, i_t), t = 1, 2, \ldots\}$.

To determine which pre-clearing system contents can ever be attained in at least one underlying state in period t, we express

$$\Psi_{(r_t)} = \bigcup_{q \in \{0,\ldots,Q\}} \{x \oplus q \in \Phi : \prod_{i \in I} r_{t-1}(x, i) = 0\}, \tag{2.19}$$

and similarly

$$\Phi_{(r_t)} = \{x \in \Phi : \prod_{i \in I} r_t(x, i) = 0\}. \tag{2.20}$$

Our models become more computationally efficient by adding "dummy" states to the respective system state spaces. More specifically, we can collect the underlying states and expand the system state spaces as

$$\Lambda_{(r_t)} \subseteq \Psi_{(r_t)} \times I, \tag{2.21}$$

and

$$\Omega_{(r_t)} \subseteq \Phi_{(r_t)} \times I, \tag{2.22}$$

for $t = 1, 2, \ldots$. In this way, we can group the system states with the same system contents together, and use $\Psi_{(r_t)}$ or $\Phi_{(r_t)}$ as the "level" set to identify the system content without needing to specify the underlying states. Using levels to group system states is a common practice in stochastic modelling, especially for the models with Markov chains, to introduce desirable structure into the system state space.

Now we shall discuss how particular properties of clearing rules affect the structure and size of the system state spaces. First, let us call a clearing rule "logical" if it satisfies the following condition.

Condition 2.2.1. *(Logical Condition) At any decision epoch $t \in T$ and underlying state $i \in I$,*

(i) $r_t(\emptyset, i) = 0$ *with certainty;*

(ii) for any $x, y \in \Phi$, *if* $r_t(x, i) = 1$, *then* $r_t(x \oplus y, i) = 1$;

(iii) for any $x \leq y \in \Phi$, *if* $r_t(x, i) = 1$, *then* $r_t(y, i) = 1$.

Part (i) is almost always true for any clearing rule; part (ii) means that if we choose to clear system content x, then for any other system contents $x \oplus y$, which can be obtained after further accumulation from x, we must also clear the system; and part (iii) is reasonable because, for the same underlying state i, if we choose to clear system content x, then we should clear any system content y that is "greater" than x.

The logical condition implies a special structure for each system state space.

Property 2.2.1. *If the clearing rule r_t in period t satisfies Condition 2.2.1, then $\Psi_{(r_t)}$ and $\Phi_{(r_t)}$ can each be mapped to a connected subtree of Φ.*

Proof. Let us first prove that the tree for $\Psi_{(r_t)}$ is actually a connected subtree for Φ. Part (i) of Condition 2.2.1 implies that $\emptyset$, which is the root node for Φ, can always be the root node for $\Psi_{(r_t)}$. Part (ii) of Condition 2.2.1 suggests that starting from $\emptyset$ and proceeding on any downward path of Φ, if a node x is not in $\Psi_{(r_t)}$, then all successors of x, i.e, $x \oplus y$, are not in $\Psi_{(r_t)}$ either. In other words, there is a "cut-off" point on each downward path of Φ, and the subtree can be built by excluding the nodes beyond those cut-off points. The subtree for $\Phi_{(r_t)}$ can be built in a similar fashion. □

For the clearing rule defined in Example 2.1.2.(iii) and $Q = 2$, the subtree corresponding to $\Phi_{(r_t)}$ is illustrated in Figure 2.3.

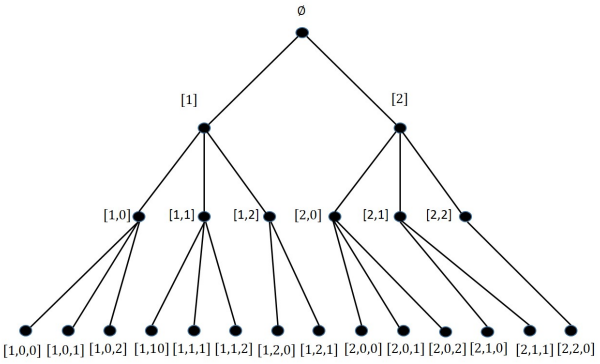

Figure 2.3: A sample tree of $\Phi_{(r_t)}$ with $Q = 2$

Theoretically, the system state space may be infinitely large if none of the clearing criteria can ever be attained. For example, some clearing processes can continue to accumulate indefinitely or up to infinite quantities. However, in practice, a clearing policy needs to be "feasible" so that the system is cleared in finite time, and the quantity cleared is often limited by a clearing capacity. For any period t, we define the maximum cumulative quantity and the oldest input age allowed by r_t, respectively, as follows:

$$\bar{Q}_{(r_t)} = \max_{(x,i)\in\Omega} \{|x| : r_t(x,i) = 0\}, \tag{2.23}$$

and

$$\bar{L}_{(r_t)} = \max_{(x,i)\in\Omega} \{\mathcal{L}(x) : r_t(x,i) = 0\}. \tag{2.24}$$

Here is the condition for a clearing rule to be feasible.

Condition 2.2.2. *(Feasibility Condition) For a given clearing rule r_t in period t, both $\bar{Q}_{(r_t)}$ and $\bar{L}_{(r_t)}$ are finite.*

The feasibility condition has a direct impact on the size of the system state space for a clearing rule.

Property 2.2.2. *In any discrete time and discrete quantity model, if the clearing rule r_t in period t satisfies Condition 2.2.2, the sets $\Psi_{(r_t)}$ and $\Phi_{(r_t)}$ are finite. However, the size of the sets grows exponentially in the order of $\mathcal{O}(Q^{\bar{L}_{(r_t)}})$.*

Proof. If $\bar{L}_{(r_t)} < \infty$, then based on equation (2.24), all sequences $x \in \Psi_{(r_t)}$ or $y \in \Phi_{(r_t)}$ must have finite lengths. Each entry in a sequence is discrete and finite according to our assumptions of discrete and finite input quantities. Since we can make all sequences into equal length by attaching zeros to the left, the total number of such sequences grows exponentially in the order of $\mathcal{O}(Q^{\bar{L}_{(r_t)}})$. □

In practice, stochastic clearing systems are often employed when the maximum delay allowed, i.e., $\bar{L}_{(r_t)}$, is relatively small. In some situations, input quantities and time periods can be approximated and discretized with larger units to make the maximum input size Q and $\bar{L}_{(r_t)}$ reasonably small without losing too much accuracy. Therefore, we argue that the insight gained by capturing more information on the system contents is enough to justify the exponential complexities of our models.

For discrete time and discrete quantity models, the finite tree structures of $\Psi_{(r_t)}$ and $\Phi_{(r_t)}$ allows us to use the *Breadth-First-Traversal* (BFT) procedures from graph theory to

construct and navigate them. The procedures for constructing $\Psi_{(r_t)}$ and $\Phi_{(r_t)}$ differ slightly and we describe them separately below.

BFT Procedure I: Constructing $\Psi_{(r_t)}$

Step 1 Initialize a list V, and store $\emptyset$ and numbers $1, 2, \ldots, Q$ in V.

Step 2 Initialize a list U, and store numbers $1, 2, \ldots, Q$ in U.

Step 3 If U is empty, go to Step 5; otherwise, read and delete the next entry from U according to the First-In-First-Out (FIFO) rule and denote it as x.

Step 4 For any underlying state $i \in I$, if $r_t(x, i) = 0$, enter $y \oplus q$ for each $q = 0, 1, \ldots, Q$ into both lists V and U; go back to Step 3.

Step 5 Output the sequences in V as $\Psi_{(r_t)}$.

BFT Procedure II: Constructing $\Phi_{(r_t)}$

Step 1 Initialize a list V, and store $\emptyset$ in V.

Step 2 Initialize a list U, and store numbers $1, 2, \ldots, Q$ in U.

Step 3 If U is empty, go to Step 5; otherwise, read and delete the next entry from U according to the First-In-First-Out (FIFO) rule and denote it as y.

Step 4 For any underlying state $i \in I$, if $r_t(y, i) = 0$, enter y to V; then for each $q = 0, 1, \ldots, Q$, enter $x \oplus q$ into U; go back to Step 3.

Step 5 Output the sequences in V as $\Phi_{(r_t)}$.

Time Complexity: Both procedures require constructing and traversing through the

entire tree only once. In Step II.4, the clearing rule r_t is checked for each underlying state $i \in I$. Therefore, according to Property 2.2.2, the time complexities of both procedures have an order of growth of $\mathcal{O}(MQ^{\bar{L}_{(r_t)}})$.

Memory Complexity: Since the entire tree needs to be stored by both procedures and each node contains a string of numbers, the memory complexities of both procedures have an order of growth of $\mathcal{O}(\bar{L}_{(r_t)}Q^{\bar{L}_{(r_t)}})$.

Please refer to Appendix C for a numerical example to verify the complexity estimates. The two BFT procedures are used in many of our algorithms, but they can be easily switched to the *Depth-First-Traversal* (DFT) procedures by using the Last-In-First-Out (LIFO) rule when reading from the list U in Step 3. The DFT Procedures have the same time and memory complexities as their counterparts.

2.2.2 Pre-Clearing System State Process

Recall that the underlying process $\{i_t, t = 1, 2, \ldots\}$ is an ergodic Markov chain, and the input process $\{(q_t, i_t), t = 1, 2, \ldots\}$ is also Markovian, i.e., q_{t+1} and i_{t+1} depend only on i_t. According to equations (2.13) and (2.15), for a bang-bang control clearing rule r_t, we have

$$x_{t+1} = x_t \cdot (1 - r_t(x_t, i_t)) \oplus q_t,$$

which only depends on (x_t, i_t), q_t, and r_t. Obviously, $\{(x_t, i_t), t = 1, 2, \ldots\}$ is a discrete time Markov chain.

There are two types of transitions for the Markov chain $\{(x_t, i_t), t = 1, 2, \ldots\}$:

(i) $(x, i) \to (x \oplus q, j)$: starting from any system state, no clearing is required at the beginning of the current period, and quantity q is received during the period

(ii) $(x, i) \to (q, j)$: starting from any system state, the system is cleared at the beginning of the current period, and quantity q is received during the period

for all $x \in \Phi$, $i, j \in I$ and $q = 0, \ldots, Q$.

The transitions for the Markov chain $\{(x_t, i_t), t = 1, 2, \ldots\}$ may be time-dependent if π is not stationary. The *one-step transition probabilities* between (x_t, i_t) and (x_{t+1}, i_{t+1}) are determined by r_t and can be written as:

$$P^{(x,r_t)}_{(x,i),(x',j)} = \begin{cases} a^{(x,r_t)}_{(x,i),(x\oplus q,j)}, & \text{if } x' = x \oplus q \\ b^{(x,r_t)}_{(x,i),(\emptyset\oplus q,j)}, & \text{if } x' = q \\ 0, & \text{otherwise} \end{cases} \tag{2.25}$$

where

$$a^{(x,r_t)}_{(x,i),(x\oplus q,j)} = (1 - r_t(x,i))[D_q]_{i,j} \tag{2.26}$$

correspond to the transition probabilities in period t without clearing, and

$$b^{(x,r_t)}_{(x,i),(q,j)} = r_t(x,i)[D_q]_{i,j} \tag{2.27}$$

are the transition probabilities in period t with clearing, and

$$\sum_{q=0}^{Q} \left(a^{(x,r_t)}_{(x,i),(x\oplus q,j)} + b^{(x,r_t)}_{(x,i),(q,j)} \right) = [D]_{i,j}, \tag{2.28}$$

for all $t = 1, 2, \ldots, N$.

We can collect the underlying states and express the transition probabilities in matrix form. More specifically, if $x' = x \oplus q$, we have

$$P^{(x,r_t)}_{x,x'} = A^{(x,r_t)}_{x,x\oplus q} = \begin{pmatrix} (1 - r_t(x,1)) & & \\ & \ddots & \\ & & (1 - r_t(x,M)) \end{pmatrix} D_q, \tag{2.29}$$

else if $x' = q$, we have

$$P_{x,x'}^{(x,r_t)} = B_{x,q}^{(x,r_t)} = \begin{pmatrix} r_t(x,1) & & \\ & \ddots & \\ & & r_t(x,M) \end{pmatrix} D_q, \tag{2.30}$$

otherwise,

$$P_{x,x'}^{(x,r_t)} = 0. \tag{2.31}$$

Together, we have

$$\sum_{q=0}^{Q} \left(A_{x,x\oplus q}^{(x,r_t)} + B_{x,q}^{(x,r_t)} \right) = D. \tag{2.32}$$

Note that $P_{x,x'}^{(x,r_t)}$ are the block matrices in $P^{(x,r_t)}$, which is the transition probability matrix for the Markov chain $\{(x_t, i_t), t = 1, 2, \ldots\}$.

For each $x \in \Phi$, the block matrices in Equations (2.29) and (2.30) can be formed by first computing $r_t(x, i)$ for all $i \in I$, and then forming the diagonal matrices and multiplying them by D_q for all $q = 0, 1, \ldots, Q$. Therefore, forming $P^{(x,r_t)}$ by blocks is more efficient than constructing it by Equations (2.25) to (2.27).

2.2.3 Post-Clearing System State Process

Now, suppose we are interested in keeping track of the post-clearing system states. According to equations (2.13) and (2.15), for bang-bang control clearing rule r_{t+1}, we have

$$y_{t+1} = x_{t+1} \cdot (1 - r_{t+1}(x_{t+1}, i_{t+1})) = (y_t \oplus q_t) \cdot (1 - r_{t+1}(y_t \oplus q_t, i_{t+1})).$$

This shows that y_{t+1} depends only on $y_t \oplus q_t$, i_{t+1}, and r_{t+1}, which then verifies the Markovian property of $\{(y_t, i_t), t = 1, 2, \ldots\}$.

There are three types of transitions for the Markov chain $\{(y_t, i_t), t = 1, 2, \ldots\}$:

(i) $(\emptyset, i) \to (\emptyset, j)$: starting from an empty system state, a positive quantity q is received and the system is cleared at the beginning of next period; or no input is received so no clearing is required at the beginning of next period;

(ii) $(y, i) \to (y \oplus q, j)$: starting from any system state, quantity q is received in the current period and no clearing is required at the beginning of next period;

(iii) $(y, i) \to (\emptyset, j)$: starting from a non-empty system state, q is the quantity received in the current period and the system is cleared at the beginning of next period;

for all $y \in \Phi$, $i, j \in I$ and $q = 0, \ldots, Q$.

The time-dependent *one-step transition probabilities* between (y_t, i_t) and (y_{t+1}, i_{t+1}) are determined by r_{t+1} and can be written as:

$$P^{(y,r_{t+1})}_{(y,i),(y',j)} = \begin{cases} a^{(y,r_{t+1})}_{(\emptyset,i),(\emptyset,j)} + b^{(y,r_{t+1})}_{(\emptyset,i),(\emptyset,j)}, & \text{if } y = y' = \emptyset \\ a^{(y,r_{t+1})}_{(y,i),(y\oplus q,j)}, & \text{if } y' = y \oplus q \neq \emptyset \\ b^{(y,r_{t+1})}_{(y,i),(\emptyset,j)}, & \text{if } y \neq \emptyset \text{ and } y' = \emptyset \\ 0, & \text{otherwise} \end{cases} \tag{2.33}$$

where

$$a^{(y,r_{t+1})}_{(y,i),(y\oplus q,j)} = (1 - r_{t+1}(y \oplus q, j))[D_q]_{i,j} \tag{2.34}$$

correspond to the transition probabilities without clearing, and

$$b^{(y,r_{t+1})}_{(y,i),(\emptyset,j)} = \sum_{q=0}^{Q} r_{t+1}(y \oplus q, j)[D_q]_{i,j} \tag{2.35}$$

are the transition probabilities with clearing, and

$$\sum_{q=0}^{Q} \left(a^{(y,r_{t+1})}_{(y,i),(y\oplus q,j)} + b^{(y,r_{t+1})}_{(y,i),(\emptyset,j)} \right) = [D]_{i,j}, \tag{2.36}$$

for all $t = 1, 2, \ldots, N$.

Expressing the transition probabilities in the matrix form, if $y = y' = \emptyset$, we have

$$\begin{aligned} P_{\emptyset,\emptyset}^{(y,r_{t+1})} &= A_{\emptyset,\emptyset}^{(y,r_{t+1})} + B_{\emptyset,\emptyset}^{(y,r_{t+1})} \\ &= D_0 + \sum_{q=1}^{Q} D_q \begin{pmatrix} r_{t+1}(q,1) & & \\ & \ddots & \\ & & r_{t+1}(q,M) \end{pmatrix}, \end{aligned} \tag{2.37}$$

else if $y' = y \oplus q$, we have

$$P_{y,y\oplus q}^{(y,r_{t+1})} = A_{y,y\oplus q}^{(y,r_{t+1})} = D_q \begin{pmatrix} (1 - r_{t+1}(y \oplus q, 1)) & & \\ & \ddots & \\ & & (1 - r_{t+1}(y \oplus q, M)) \end{pmatrix}, \tag{2.38}$$

else if $y' = \emptyset$, we then have

$$P_{y,\emptyset}^{(y,r_{t+1})} = B_{y,\emptyset}^{(y,r_{t+1})} = \sum_{q=0}^{Q} D_q \begin{pmatrix} r_{t+1}(y \oplus q, 1) & & \\ & \ddots & \\ & & r_{t+1}(y \oplus q, M) \end{pmatrix}, \tag{2.39}$$

otherwise,

$$P_{y,y'}^{(y,r_{t+1})} = 0. \tag{2.40}$$

Together, we have

$$\sum_{q=0}^{Q} \left(A_{y,y\oplus q}^{(y,r_t)} + B_{y,\emptyset}^{(y,r_t)} \right) = D. \tag{2.41}$$

Once again, we note that $P_{y,y'}^{(y,r_t)}$ are the block matrices in $P^{(y,r_t)}$, which is the transition probability matrix for the Markov chain $\{(y_t, i_t), t = 1, 2, \ldots\}$.

In the context of a stationary policy for the infinite horizon problem, we can drop the subscript t from all of the above definitions, and the transition probabilities in matrices $P^{(x,r)}$ and $P^{(y,r)}$ are independent of time. Therefore, the Markov chains $\{(x_t, i_t), t =$

$1, 2, \ldots\}$ and $\{(y_t, i_t), t = 1, 2, \ldots\}$ are both time-homogeneous.

Theorem 2.2.3. *For discrete-quantity problems, assume that $D_0\boldsymbol{e} \neq \boldsymbol{e}$ and D is irreducible. Under a logical, feasible, and stationary policy $\pi = (r, r, \ldots)$, the processes $\{(x_t, i_t), t = 1, 2, \ldots\}$ and $\{(y_t, i_t), t = 1, 2, \ldots\}$ are both ergodic Markov chains with finite state space $\Lambda_{(r)}$ and $\Omega_{(r)}$ with a tree structure, respectively.*

Proof. First, since the clearing policy is stationary, the transitions of the Markov chain $\{(x_t, i_t), t = 1, 2, \ldots\}$ are all within the system state space $\Lambda_{(r)}$ and the transition probability matrix $P^{(x,r)}$ is time-independent. From the definitions of the input process and the state transitions, if the policy satisfies Condition 2.2.1, it is easy to verify that the states in $\Lambda_{(r)}$ communicate with one another and are positive recurrent. This is because after each clearing, the system state is set to $(\emptyset, j)$ for some $j \in I$, and another clearing cycle begins. Since $\{i_t, t = 1, 2, \ldots\}$ is irreducible, $\{(x_t, i_t), t = 1, 2, \ldots\}$ is also irreducible. Property 2.2.2 guarantees that $\Lambda_{(r)}$ is finite. Hence, $\{(x_t, i_t), t = 1, 2, \ldots\}$ is positive recurrent. Property 2.2.1 confirms the tree structures of the state spaces. The proof is essentially the same for $\{(y_t, i_t), t = 1, 2, \ldots\}$. □

Note: The definitions for the transition probabilities in Equations (2.25) to (2.28) and Equations (2.33) to (2.36) are interchangeable, depending on which Markov chain we are referring to.

2.3 Cost Structure

Three types of costs are relevant to our analysis of stochastic clearing systems: (i) *delay penalty cost*, (ii) *fixed clearing cost*, and (iii) *variable clearing cost*.

2.3.1 Delay Penalty Cost

The delay penalty cost, also known as the disutility of waiting, or the holding cost in many practical applications, is incurred during every period that an input remains in the system. In earlier works on stochastic clearing systems, more specifically, in inventory theory and shipment consolidation models, the delay penalty cost was charged at a constant rate over time. In other words, the total delay penalty over the clearing cycle was assumed to be linear with respect to the length of time that input was in the system. However, we argue that it is more realistic for the penalty rate to actually "grow" as the delay is prolonged, i.e., as each input "ages". Therefore, in our models, the total delay penalty cost in a clearing cycle is non-decreasing with respect to the length of time each input was in the system. This modification of the delay penalty cost function is among the major contributions of our research.

Recording both the quantities and ages of individual outstanding inputs in the system content variables x_t and y_t becomes necessary when the delay penalty rates vary over time and the penalties for individual inputs are additive or super-additive. Even though the delay penalties are usually only charged upon clearing, we assume that they are incurred in each period throughout the clearing cycle and summed to the final amount. Therefore, at each decision epoch t, we assume that the delay penalty cost is calculated by a function $H_t : \Phi \to \mathbb{R}$, which is a function of the post-clearing system content variable y_t.

The exact form of H_t may vary depending on the actual calculations of the delay penalties. Some typical examples are shown below.

Example 2.3.1. *Suppose that* $y_t = \{y_{[l]}, \ldots, y_{[1]}\}$

(a) $H_t(y_t) = 0.1|y_t|^2$

(b) $H_t(y_t) = 0.1\mathcal{L}(y_t)^2$

(c) $H_t(y_t) = \sum_{j=1}^{l} 0.1 \left(j \cdot \delta_{\{[y_{[j]}]>0\}}\right)^2$

(d) $H_t(y_t) = \sum_{j=1}^{l} 0.1 \left(j \cdot y_{[j]}\right)^2$

where $\delta_{\{.\}}$ *is the indicator function.*

The penalty function in (a) is a polynomial in the total cumulative quantities; similarly, (b) is polynomial in the maximum age of all outstanding inputs; on the other hand, the function in (c) is the sum of polynomials of individual input ages; and lastly, part (d) is a sum of products between input quantities, ages, and penalty rates.

2.3.2 Clearing Costs

Recall that w_t is the content cleared in period t and there are two types of clearing costs: fixed clearing cost and variable clearing cost. For an arbitrary period t, the fixed clearing cost in the period is given by the function $K_t : \Phi \to \mathbb{R}_{\geq 0}$, and generally has the form

$$K_t(w_t) = \hat{k}_t \cdot \delta_{\{|w_t|>0\}}, \tag{2.42}$$

where $w_t = x_t - y_t$ and $\hat{k}_t$ is a constant in every decision epoch $t \in T$. This means that the fixed clearing cost is only charged when a positive quantity is cleared, but that cost itself may change over time.

The variable clearing cost function, $C_t : \Phi \to \mathbb{R}$, is usually calculated as a linear function of the cleared quantities, but in some cases, quantity discounts can be obtained for larger clearings. In later chapters, we show that this cost has little impact on the choice of optimal clearing policy, unless there is a significant quantity discount or monetary discount over time.

Example 2.3.2. *Sample variable clearing cost functions*

(a) $C_t(w_t) = \hat{c}_t \cdot |w_t|$

(b) $C_t(w_t) = \hat{c}_t \cdot \dfrac{|w_t|}{|w_t| + a}$

(c) $C_t(w_t) = \begin{cases} \hat{c}^N |w_t|, & \text{if } |w_t| < \bar{W} \\ \hat{c}^V \bar{W}, & \text{if } \bar{W} \leq |w_t| \leq \hat{W} \\ \hat{c}^V |w_t|, & \text{if } \hat{W} < |w_t| \end{cases}$

where $\hat{c}_t, \forall t \in T$, $\hat{c}^N$, *and* $\hat{c}^V$ *are variable cost rates, and* $\bar{W}$ *and* $\hat{W}$ *are quantity thresholds to qualify for the discounted rate.*

2.3.3 Objective Cost Functions

Minimizing the total costs associated with a stochastic clearing system is one of the main objectives of our research. We use the variations of

$$\mathcal{C}^{\pi,\mathrm{T},\alpha}_{<t_1,t_2>}(x,i) \tag{2.43}$$

where (x, i) is the initial system state, to denote different cost objectives that we are interested in. The subscripts and superscripts in the above notation are used to distinguish the different objectives we may have. For instance, the two numbers $1 \leq t_1 \leq t_2 \leq N$ indicate the time interval over which we are computing costs. $0 \leq \alpha \leq 1$ is the discount factor, for which $\alpha = 0$ means only the cost in the current period is captured, and $\alpha = 1$ means no discount is applied. In that case, we can drop the subscript $< t_1, t_2 >$. The clearing policy is given by π, and we can substitute π by r if the policy is stationary. Superscript "T" signals that we are computing the total cost over the time interval, and we can substitute it by "A" if we are interested in the average cost over the time interval instead. Lastly, the domain of the above function is the system state space Ω, where $(x, i) \in \Omega$ indicates the initial system state at the beginning of period t_1.

Utilizing the above notation, the *expected total cost* over the planning horizon is defined

as

$$\begin{aligned}&\mathcal{C}^{\pi,\mathrm{T}}_{<1,N>}(x,i)\\&=E\left[\sum_{t=1}^{N}\left(H_t(y_t)+K_t(w_t)+C_t(w_t)\right)+K_{N+1}(x_{N+1})+C_{N+1}(x_{N+1})\middle|(x_1,i_1)=(x,i)\right].\end{aligned}\tag{2.44}$$

Similarly, the *expected total discounted cost* over the planning horizon is calculated as

$$\begin{aligned}&\mathcal{C}^{\pi,\mathrm{T},\alpha}_{<1,N>}(x,i)\\&=E\left[\sum_{t=1}^{N}\alpha^{t-1}\left(H_t(y_t)+K_t(w_t)+C_t(w_t)\right)+\alpha^{N}(K_{N+1}(x_{N+1})+C_{N+1}(x_{N+1}))\middle|(x_1,i_1)=(x,i)\right],\end{aligned}\tag{2.45}$$

for some $\alpha<1$. Finally, the *expected average total cost per period over* over the planning horizon can be computed as

$$\mathcal{C}^{\pi,\mathrm{A}}_{<1,N>}(x,i)=\frac{\mathcal{C}^{\pi,\mathrm{T}}_{<1,N>}(x,i)}{N}.\tag{2.46}$$

For infinite horizon problems, we would take the limit of the above equations as $N\to\infty$.

Chapter 3

Expected Total Cost Model over Finite Horizon

In this chapter, we study stochastic clearing problems over a finite planning horizon with the expected total cost as our objective function. We first present an algorithm to compute the expected total cost for any given clearing policy. Then we construct algorithms to find the *optimal* clearing policies that minimize the expected total cost. Special structures and characteristics of the optimal policies are identified under certain scenarios. The modelling assumptions and notations used in this chapter are summarized below.

(i) The planning horizon is finite, i.e., $T = \{1, 2, \ldots, N\}$, where $N < \infty$.

(ii) The model is in discrete time and assumes discrete quantities, and the input process is modelled as a *BMAP*.

(iii) The clearing policy is given as $\pi = (r_1, r_2, \ldots, r_N)$.

(iv) The pre-clearing system state process $\{(x_t, i_t), t = 1, 2, \ldots\}$ is the Markov chain of interest here.

(v) The cost functions are given as H_t, K_t, and C_t, which may vary over time.

3.1 Policy Evaluation

For any given clearing policy, the expected total cost over the finite horizon can be computed using backward induction. In order to use such a method, we need to specify the "reward" functions for each action selected for each system state $(x, i) \in \Lambda_{(r_t)}$ in period t. We define the reward function as

$$u_t^{\pi}(x, i) = H_t(y) + K_t(x - y) + C_t(x - y), \tag{3.1}$$

where the post-clearing system content y is determined by $r_t(x, i)$ in period t. If no partial clearing is allowed, we can rewrite the reward function as

$$u_t^{\pi}(x, i) = H_t((1 - r_t(x, i))x) + K_t(r_t(x, i)x) + C_t(r_t(x, i)x), \tag{3.2}$$

where $r_t(x, i) = 0$ means continuing to accumulate and $r_t(x, i) = 1$ means clearing the system.

Subject to the clearing policy π, at any arbitrary decision epoch n, we define the expected total cost starting from period n until the end of the planning horizon as

$$\begin{aligned}
&\mathcal{C}^{\pi,\mathrm{T}}_{<n,N>}(x, i) \\
&= E\left[\sum_{t=n}^{N} \left(H_t(y_t) + K_t(x_t - y_t) + C_t(x_t - y_t)\right) + K_{N+1}(x_{N+1}) + C_{N+1}(x_{N+1})\Big|(x_n, i_n) = (x, i)\right] \\
&= E\left[\sum_{t=n}^{N} u_t^{\pi}(x_t, i_t) + K_{N+1}(x_{N+1}) + C_{N+1}(x_{N+1})\Big|(x_n, i_n) = (x, i)\right],
\end{aligned} \tag{3.3}$$

for all $n = 1, 2, \ldots, N$, and

$$\mathcal{C}^{\pi,\mathrm{T}}_{<N+1,N>}(x, i) = K_{N+1}(x) + C_{N+1}(x). \tag{3.4}$$

For any given initial system state (x, i) in period n, the Markovian property of $\{(x_t, i_t), t =$

$1, 2, \ldots\}$ allows us to write $\mathcal{C}^{\pi,\mathrm{T}}_{<n,N>}(x,i)$ recursively as

$$\mathcal{C}^{\pi,\mathrm{T}}_{<n,N>}(x,i) = u^{\pi}_{n}(x,i) + E\left[\mathcal{C}^{\pi,\mathrm{T}}_{<n+1,N>}(x',j)\middle|(x,i)\right], \tag{3.5}$$

which can be computed as

$$\mathcal{C}^{\pi,\mathrm{T}}_{<n,N>}(x,i) = u^{\pi}_{n}(x,i) + \sum_{(x',j)\in\Lambda_{(r_n)}} P^{(x,r_n)}_{(x,i),(x',j)}\mathcal{C}^{\pi,\mathrm{T}}_{<n+1,N>}(x',j), \tag{3.6}$$

where the transition probabilities are given by Equations (2.25) to (2.28).

We now present an algorithm to compute the expected total cost over a finite horizon.

Algorithm I: Finite Horizon Policy Evaluation Algorithm

I.1 Construct $\Psi_{(r_{N+1})}$ using BFT Procedure I. For each $(x,i) \in \Psi_{(r_{N+1})} \times I$, compute $\mathcal{C}^{\pi,\mathrm{T}}_{<N+1,N>}(x,i)$ by Equation (3.4). Set $n := N$.

I.2 Construct $\Psi_{(r_n)}$ using the BFT Procedure I. For each $(x,i) \in \Psi_{(r_n)} \times I$, compute $\mathcal{C}^{\pi,\mathrm{T}}_{<n,N>}(x,i)$ by Equations (2.25), (3.2), and (3.6).

I.3 If $n = 1$, stop and report $\mathcal{C}^{\pi,\mathrm{T}}_{<1,N>}(x,i)$; otherwise, set $n := n - 1$ and return to Step I.2.

Time Complexity: The time complexity of BFT Procedure I has an order of growth of $\mathcal{O}(MQ^{\bar{L}_{(r_t)}})$, for each $t \in T$. Step I.2 is repeated N times and in the n-th iteration, it must scan through both $\Psi_{(r_{n-1})} \times I$ and $\Psi_{(r_n)} \times I$. This step has an order of growth of $\mathcal{O}(M^2Q^{\bar{L}_{(r_t)}})$ in a single iteration. Therefore, the total time complexity of Algorithm I is $\mathcal{O}(M^2NQ^{\bar{L}_{(r_t)}})$.

Memory Complexity: All sequences in $\Psi_{(r_{n-1})}$ and $\Psi_{(r_n)}$ must be stored in iteration n. However, the rewards and transition probabilities for each sequence do not need to be

stored. Therefore, the memory complexity is $\mathcal{O}(\bar{L}_{(r_t)}Q^{\bar{L}_{(r_t)}})$.

Please refer to Appendix C for a numerical example to verify the complexity estimates. We demonstrate Algorithm I through the next numerical example.

Example 3.1.1. *Consider a stochastic clearing problem with the following parameters.*

(i) The planning horizon has ten periods, i.e., $N = 10$.

(ii) The input process is described in Example 2.1.1.(ii).

(iii) The delay penalty cost functions H_t *is described in Example 2.3.1.(d).*

(iv) The fixed clearing cost is given as $K_t(w_t) = 10 \cdot \delta_{\{|w_t|>0\}}$.

(v) The variable clearing cost is given as $C_t(w_t) = 0.5|w_t|$.

(vi) The clearing policy is given as

$$r_t(x_t, i_t) = \begin{cases} 0, & \text{if } |x_t| < 5 \text{ and } \mathcal{L}(x_t) < 4 \\ 1, & \text{otherwise} \end{cases}, \text{ for } t \leq 5;$$

$$r_t(x_t, i_t) = \begin{cases} 0, & \text{if } |x_t| < 4 \text{ and } \mathcal{L}(x_t) < 3 \\ 1, & \text{otherwise} \end{cases}, \text{ for } 5 < t \leq 10.$$

The expected total costs for this particular problem is computed using Algorithm I and summarized in Table 3.1.

From the preceding numerical example, we can make the following observations. First, we notice that the input process is a Markov Modulated Input Process, so the input arrival rate may vary depending on the underlying state. Therefore, we can see that for every sequence x in Table 3.1, $\mathcal{C}^{\pi,\mathrm{T}}_{<1,N>}(x, 1) \neq \mathcal{C}^{\pi,\mathrm{T}}_{<1,N>}(x, 2)$. Thus, the initial underlying state affects the expected total cost.

$\mathcal{C}^{\pi,\mathrm{T}}_{<1,N>}(x,i)$	$i=1$	$i=2$
$\emptyset$	5.3235	3.4467
$x=[1]$	8.7308	6.6800
$x=[2]$	9.1352	7.0717
$x=[1,0]$	9.4345	7.6600
$x=[1,1]$	10.0845	8.3100
$x=[1,2]$	10.5345	8.7600
$x=[2,0]$	9.8845	8.1100
$x=[2,1]$	10.5345	8.7600
$x=[2,2]$	10.9845	9.2100
$x=[1,0,0]$	9.8235	7.9467
$x=[1,0,1]$	10.3235	8.4467
$x=[1,0,2]$	10.8235	8.9467
$x=[1,1,0]$	10.3235	8.4467
$x=[1,1,1]$	10.8235	8.9467
$x=[1,1,2]$	11.3235	9.4467
$x=[1,2,0]$	10.8235	8.9467
$x=[1,2,1]$	11.3235	9.4467
$x=[1,2,2]$	11.8235	9.9467
$x=[2,0,0]$	10.3235	8.4467
$x=[2,0,1]$	10.8235	8.9467
$x=[2,0,2]$	11.3235	9.4467
$x=[2,1,0]$	10.8235	8.9467
$x=[2,1,1]$	11.3235	9.4467
$x=[2,1,2]$	11.8235	9.9467
$x=[2,2,0]$	11.3235	9.4467
$x=[2,2,1]$	11.8235	9.9467
$x=[2,2,2]$	12.3235	10.4467

Table 3.1: Summary of Expected Total Costs for Example 3.1.1

Next, we see that the delay penalty function is the sum of products between input quantities, ages, and penalty rates. Therefore, the input sequence of the initial system content has a direct impact on the expected total cost if no dispatch is required immediately. For example, when $x=[1,0]$ or $x=[2]$, no dispatch is required. We then see that $\mathcal{C}^{\pi,\mathrm{T}}_{<1,N>}([1,0],i) > \mathcal{C}^{\pi,\mathrm{T}}_{<1,N>}([2],i)$, for each $i=1,2$. These results mean that one unit of input that has already stayed in the system for two periods will incur more cost than two units of input that has just arrived in the last period.

On the other hand, for $x=[1,2,2]$, $x=[2,1,2]$, and $x=[2,2,1]$, since dispatch is

required immediately for all three cases, the difference in the input sequences no longer matters as the total accumulated quantities are the same. Therefore, $\mathcal{C}^{\pi,\mathrm{T}}_{<1,N>}([1,2,2],i) = \mathcal{C}^{\pi,\mathrm{T}}_{<1,N>}([2,1,2],i) = \mathcal{C}^{\pi,\mathrm{T}}_{<1,N>}([2,2,1],i)$, for each $i = 1, 2$. These observations generally hold true and they can be used to characterize the optimal clearing policy.

3.2 Optimality Equations

One of our main goals in this chapter is to find an optimal deterministic clearing policy, denoted by $\pi^* = (r_1^*, r_2^*, \ldots, r_N^*)$, that minimizes the expected total cost during the entire planning horizon, if the initial system state is (x, i). Recall from Section 3.1, if such optimal policy exists, then the minimum expected total cost $\mathcal{C}^{\pi^*,\mathrm{T}}_{<1,N>}(x,i)$, defined by Equation (3.3), must satisfy

$$\mathcal{C}^{\pi^*,\mathrm{T}}_{<1,N>}(x,i) \leq \mathcal{C}^{\pi,\mathrm{T}}_{<1,N>}(x,i), \forall \pi \in \Pi. \tag{3.7}$$

We shall call the stochastic clearing problem over a finite sub-interval $< n, N >$ the *last (N-n)-stage* clearing problem, and denote the corresponding sub-policy by $\pi_n = (r_n, r_{n+1}, \ldots, r_N)$ and the expected total cost function by $\mathcal{C}^{\pi,\mathrm{T}}_{<n,N>}(x,i)$.

For any stochastic clearing system over a finite horizon, we can establish the existence of an optimal clearing policy if the cost functions satisfy the following conditions.

Condition 3.2.1.

(i) $K_t(w) = \hat{k}_t \cdot \delta_{\{|w|>0\}}$ *and* $\hat{k}_t \geq \hat{k}_{t+1} \geq \ldots \geq \hat{k}_{N+1} = 0, \forall t \in T$, *i.e., the fixed clearing cost is non-increasing over time and is reduced to zero at the end of the horizon.*

(ii) $C_t(w) = \hat{c}_t \cdot |w|$, *i.e., the variable clearing cost is a linear function of the cleared quantities.*

(iii) $H_t(x+y) \geq H_t(x) + H_t(y)$, *for all* $x, y \in \Phi$ *and* $t \in T$, *i.e.,* H_t *is superadditive in* Φ.

(iv) $H_t(x \oplus [0,\ldots,0] \oplus z) \geq H_t(x \oplus z)$, *for all* $x, y, z \in \Phi$ *and* $t \in T$, *i.e., delay penalties are non-decreasing over time.*

(v) $H_t(x) - C_t(x) + C_{t+1}(x) \geq 0$, *for all* $x \in \Phi$ *and* $t \in T$, *i.e., The variable clearing cost reduction from delaying clearance does not exceed the corresponding delay penalty.*

(vi) *For any real number* $M > 0$, *there are only finitely many sequences* $x \in \Phi$ *such that* $H_t(x) - C_t(x) + C_{t+1}(x) \leq M$, *for all* $t \in T$.

(vii) $H_t(x) \geq H_t(\emptyset) = 0$, *for all* $x \in \Phi$ *and* $t \in T$.

Part (i) is required to ensure that there is no incentive to clear the system immediately due to lower fixed clearing cost at the present moment than in the future. Part (ii) simplifies the variable clearing cost function to a linear function for any non-zero quantities, so that $C_t(x+y) = C_t(x)+C_t(y)$ and $C_t(x \oplus y) = C_t(x)+C_t(y)$, for all $x, y \in \Phi$. Part (iii) suggests that the delay penalty function is superadditive which implies that the delay penalty gets more expensive as more input quantities and input ages are accumulated. Part (iv) is an essential part of our modelling assumption because it accounts for the ages of inputs in each sequence. Part (v) ensures that if we delay the clearing of any input sequence for an extra period, the variable cost savings cannot exceed the delay penalty incurred by that sequence. In other words, the reduction of variable clearing cost does not provide sufficient incentive for us to continue accumulating. In fact, the clearing decisions have very little impact on the variable clearing costs because all inputs will eventually be cleared. Part (vi) is necessary so that clearing is guaranteed eventually. The impact of this part will be explained later. Lastly, part (vii) indicates that no delay penalty is incurred if the system is empty.

Note: Condition 3.2.1 cannot guarantee that $H_t(x) \geq H_t(y)$ if $x \not\leq y$. In other words, if the sequence cannot be ranked with the partial order given in Definition 2.1.2, then we must compute and compare the actual delay penalties. This is mainly because both the

quantity and the age of each outstanding input contribute to the delay penalty at the same time, hence their effects are "intertwined". For instance, an input of 10 units which has been delayed for 5 periods may lead to a higher penalty than an input of 20 units that has only been delayed for 2 periods.

Condition 3.2.1 allows us to rewrite Equation (3.3) into more manageable forms for the purpose of our analysis. First, we can drop the term $K_{N+1}(\cdot)$ because of part (i). Recall that $x_{t+1} = y_t \oplus q_t$, so together with Condition 3.2.1.(ii), we have

$$\mathcal{C}^{\pi,\mathrm{T}}_{<n,N>}(x,i) = E\left[\sum_{t=n}^{N}\left(K_t(w_t) + H_t(y_t) - C_t(y_t) + C_{t+1}(y_t \oplus q_t)\right)\Big|(x_n,i_n) = (x,i)\right] + C_n(x), \tag{3.8}$$

where q_t is the input quantity in period t. Because $C_n(x)$ does not depend on π, we can drop that term and modify the function in (3.8) to be

$$\mathcal{C}^{\pi,\mathrm{T}}_{<n,N>}(x,i) = E\left[\sum_{t=n}^{N}\left(K_t(w_t) + H_t(y_t) - C_t(y_t) + C_{t+1}(y_t \oplus q_t)\right)\Big|(x_n,i_n) = (x,i)\right]. \tag{3.9}$$

Define a function G_t as

$$\begin{aligned} G_t(y,i) &= H_t(y) - C_t(y) + \sum_{q=0}^{Q} C_{t+1}(y \oplus q) f_i(q) \\ &= H_t(y) - C_t(y) + C_{t+1}(y) + \sum_{q=0}^{Q}\sum_{j=1}^{M} C_{t+1}(q)[D_q]_{i,j}. \end{aligned} \tag{3.10}$$

$G_t(y,i)$ is equivalent to the delay penalty for y in period t plus the expected increment in variable clearing cost between y in period t and $y \oplus q$ in period $t+1$. Condition 3.2.1.(v) ensures that $G_t(y,i) \geq 0$, for all $y \in \Phi$ and $t \in T$. If we substitute (3.10) into (3.9), we have

$$\mathcal{C}^{\pi,\mathrm{T}}_{<n,N>}(x,i) = \sum_{t=n}^{N} E\left[K_t(w_t) + G_t(y_t,i_t)\Big|(x_n,i_n) = (x,i)\right]. \tag{3.11}$$

To further simplify our notations, let us define an operator

$$\begin{aligned} \mathcal{E}b(y,i) &= E\left[b(y \oplus q, j)|i\right] \\ &= \sum_{j=1}^{M} \sum_{q=0}^{Q} b(y \oplus q, j)[D_q]_{i,j}, \end{aligned} \tag{3.12}$$

where $b(y,i) : \Phi \times I \to \mathbb{R}$. We can interpret $\mathcal{E}b(y,i)$ as the expected value of function b after the input is received in the period, given that at the beginning of the period the system state is (y,i).

The clearing policy π is said to be *optimal* if the corresponding $(N-n)$-stage policies satisfy the following optimality equations

$$\mathcal{C}^{\pi,\mathrm{T}}_{<n,N>}(x,i) = \inf_{y \le x} \left\{ K_n(x-y) + G_n(y,i) + \mathcal{E}\mathcal{C}^{\pi,\mathrm{T}}_{<n+1,N>}(y,i) \right\}, \tag{3.13}$$

for $t = 1, 2, \ldots, N$. At the end of the planning horizon, since $\hat{k}_{N+1} = 0$ and C_{N+1} is already accounted for by G_N, we can modify Equation (3.4) to

$$\mathcal{C}^{\pi,\mathrm{T}}_{<N+1,N>}(x,i) = 0. \tag{3.14}$$

A solution to the system of Equations (3.13) and boundary condition (3.14) is a sequence of functions $\mathcal{C}_n : \Lambda_{(r_t)} \to \mathbb{R}, n = 1, 2, \ldots, N+1$, with the property that $\mathcal{C}_{N+1}$ satisfies Equation (3.14), $\mathcal{C}_N$ satisfies the N-th equation with $\mathcal{C}_{N+1}$ substituted into the RHS of the N-th equation and so forth. Hence, the solution can be obtained via backward induction.

Now, we define the value function for the problem over the time interval $< n, N >$, with initial system state $(x_n, i_n) = (x,i)$, to be

$$V_n(x,i) = \inf_{\pi \in \Pi} \mathcal{C}^{\pi,\mathrm{T}}_{<n,N>}(x,i). \tag{3.15}$$

Note that the existence of an optimal policy is not required to define the value function.

Of course, once the existence is established, the "inf" in (3.15) can be replaced by "min".

Using the principle of optimality, we can write the following dynamic programming equations for the value function

$$\begin{aligned} V_n(x,i) &= \inf_{y \le x} \{K_n(x-y) + G_n(y,i) + E[V_{n+1}(y \oplus q_n, i_{n+1})|i_n = i]\} \\ &= \inf_{y \le x} \{K_n(x-y) + G_n(y,i) + \mathcal{E}V_{n+1}(y,i)\}, \forall n \in T, \\ V_{N+1}(x,i) &= 0. \end{aligned} \tag{3.16}$$

If we let

$$W_n(y,i) = G_n(y,i) + \mathcal{E}V_{n+1}(y,i), \forall n \in T, \tag{3.17}$$

we can rewrite the dynamic programming Equations (3.16) as

$$\begin{aligned} V_n(x,i) &= \inf_{y \le x} \{K_n(x-y) + W_n(y,i)\}, \forall n \in T, \\ V_{N+1}(x,i) &= 0. \end{aligned} \tag{3.18}$$

3.3 Characterizing the Optimal Policies

In this section, we show some characteristics of the optimal clearing policies for the finite-horizon clearing problems. All of the results obtained are very intuitive and they confirm some basic assumptions about the clearing processes. Since the system state process $\{(x_t, i_t), t = 1, 2, \ldots\}$ is a Markov chain, Markovian clearing rules are sufficient to determine the best action for the future. If the current system state is completely observable, any optimal policy can determine the best action based on the system state. Thus, an optimal clearing policy can be Markovian and deterministic (MD).

3.3.1 Effects of Initial System State

In finite horizon problems, the initial system state plays a significant role in determining the expected total cost. For instance, if the planning horizon begins with some outstanding inputs already in the system as opposed to an empty system, then we would expect the total cost to be higher. The following lemmas use the dynamic programming equations defined in Section 3.2 to show the effects of initial system state on the value function.

Lemma 3.3.1. *Under Condition 3.2.1, for any pair of system states $x, x' \in \Phi$ such that $x' \leq x$, we have*

$$G_t(x', i) \leq G_t(x, i). \tag{3.19}$$

for all $i \in I$ and $t \in T$.

Proof. Since $x' \leq x$, let $z = x - x' \in \Phi$. We have

$$\begin{aligned}
&G_t(x, i) - G_t(x', i) \\
&= -\left[C_t(x) - C_t(x')\right] + \left[H_t(x) - H_t(x')\right] + \sum_{j=1}^{M} \sum_{q=0}^{Q} \left[C_{t+1}(x \oplus q) - C_{t+1}(x' \oplus q)\right] [D_q]_{i,j} \\
&\geq -C_t(z) + H_t(z) + C_{t+1}(z) \geq 0,
\end{aligned}$$

The first inequality follows from parts (ii) and (iii) of Condition 3.2.1, and the fact that $(z \oplus 0) = (x \oplus q) - (x' \oplus q)$ and $|z \oplus 0| = |z|$. The second inequality is a direct result of part (v). □

Lemma 3.3.1 immediately suggests that $G_t(\emptyset, i) = \inf_{x \in \Phi} G_t(x, i)$ for all $i \in I$. The following two lemmas compare the value functions for a pair of sequences of initial system states.

Lemma 3.3.2. *Under Condition 3.2.1, $x' \leq x$ implies*

$$V_t(x', i) \leq V_t(x, i), \tag{3.20}$$

and

$$0 \leq G_t(x,i) - G_t(x',i) \leq W_t(x,i) - W_t(x',i), \tag{3.21}$$

for all $x, y \in \Phi$, $i \in I$ *and* $t \in T$.

Proof. We first observe that according to Condition 3.2.1 and Lemma 3.3.1, $x' \leq x$ implies that $K_t(x') \leq K_t(x)$ and $G_t(x',i) \leq G_t(x,i)$, for all $t \in T$ and $i \in I$. We then proceed with a proof by induction on t. Note that for any $i \in I$, $V_{N+1}(x',i) = V_{N+1}(x,i) = 0$, then

$$\begin{aligned} W_N(x,i) - W_N(x',i) &= G_N(x,i) - G_N(x',i) + \mathcal{E}V_{N+1}(x,i) - \mathcal{E}V_{N+1}(x',i) \\ &= G_N(x,i) - G_N(x',i) \geq 0. \end{aligned}$$

More importantly, the above inequality suggests that $W_N(\emptyset,i) = \inf_{x\in\Phi} W_N(x,i)$. From Equation (3.18) we have

$$\begin{aligned} V_N(x',i) &= \inf_{y' \leq x'} \{K_N(x'-y') + W_N(y',i)\} \\ &= \min\{K_N(x') + W_N(\emptyset,i),\ W_N(x',i)\} \\ &\leq \min\{K_N(x) + W_N(\emptyset,i),\ W_N(x,i)\} = V_N(x,i). \end{aligned}$$

Now suppose that Equations (3.20) and (3.21) both hold for all $t \geq l+1$, we see that for $t = l$, we have

$$\begin{aligned} \mathcal{E}V_{t+1}(x,i) - \mathcal{E}V_{t+1}(x',i) &= E\left[V_{t+1}(x \oplus q_t, i_{t+1}) - V_{t+1}(x' \oplus q_t, i_{t+1}) | i_t = i\right] \\ &= \sum_{j=1}^{M}\sum_{q=0}^{Q}[V_{t+1}(x \oplus q, j) - V_{t+1}(x' \oplus q, j)][D_q]_{i,j} \\ &\geq 0, \end{aligned}$$

since $x' \leq x$ implies $x' \oplus q \leq x \oplus q$. Then we have

$$\begin{aligned} W_t(x,i) - W_t(x',i) &= G_t(x,i) - G_t(x',i) + \mathcal{E}V_{t+1}(x,i) - \mathcal{E}V_{t+1}(x',i) \\ &\geq G_t(x,i) - G_t(x',i) \geq 0. \end{aligned}$$

Lastly, since $W_t(\emptyset, i) = \inf_{x \in \Phi} W_t(x, i)$, we must have

$$\begin{aligned} V_t(x', i) &= \inf_{y' \leq x'} \{K_t(x' - y') + W_t(y', i)\} \\ &= \min \{K_t(x') + W_t(\emptyset, i),\ W_t(x', i)\} \\ &\leq \min \{K_t(x) + W_t(\emptyset, i),\ W_t(x, i)\} = V_t(x, i). \end{aligned}$$

This completes our proof. □

The proof of Lemma 3.3.2 is purely analytic. An alternative proof of inequality (3.20) may be constructed using the following argument, which may be made rigorous. Suppose the initial system state at the beginning of period t before clearance is (x', i). If we mimic the optimal clearing policy for each period $l \geq t$ as if the initial system state before dispatch is (x, i), where $x' \leq x$, then this policy incurs expected total cost which is no greater than $V_t(x, i)$. On the other hand, this policy must incur expected total cost at least as large as $V_t(x', i)$. Combining these remarks proves inequality (3.20). The inequality (3.21) can be justified in a similar way.

The following theorem is a direct result of Lemma 3.3.2.

Theorem 3.3.3. *Under Condition 3.2.1, an optimal clearing policy exists and must be a on-off control clearing policy, i.e., if a clearing is triggered then all accumulated inputs must be cleared, such that*

$$V_t(x, i) = \min \{K_t(x) + W_t(\emptyset, i), W_t(x, i)\}, \forall t \in T. \tag{3.22}$$

Proof. From Lemma 3.3.2, we have already shown that

$$V_t(x, i) = \inf_{y \leq x} \{K_t(x - y) + W_t(y, i)\} = \min \{K_t(x) + W_t(\emptyset, i), W_t(x, i)\}, \forall t \in T,$$

The existence of the value function $V_t(x, i)$ over a finite planning horizon implies the existence of an optimal clearing policy; and Equation (3.22) suggests that such policy is a

on-off control policy. □

Lemma 3.3.4. *Under Condition 3.2.1, $x' \leq x$ implies*

$$V_t(x,i) \leq V_t(x',i) + \hat{k}_t, \tag{3.23}$$

and

$$W_t(x,i) - W_t(x',i) \leq G_t(x,i) - G_t(x',i) + \hat{k}_{t+1}, \tag{3.24}$$

for all $x, y \in \Phi$, $i \in I$ and $t \in T$.

Proof. According to Lemma 3.3.2, we have

$$G_t(x',i) \leq G_t(x,i),$$

and

$$W_t(\emptyset,i) \leq W_t(x',i) \leq W_t(x,i), \forall t \in T, \forall i \in I.$$

Therefore

$$\begin{aligned} V_t(x,i) &= \min\{K_t(x) + W_t(\emptyset,i),\ W_t(x,i)\} \\ &\leq \min\left\{\hat{k}_t + W_t(\emptyset,i),\ \hat{k}_t + W_t(x,i)\right\} \\ &= \min\{W_t(\emptyset,i),\ W_t(x,i)\} + \hat{k}_t \\ &= W_t(\emptyset,i) + \hat{k}_t \\ &\leq \min\{K_t(x') + W_t(\emptyset,i),\ W_t(x',i)\} + \hat{k}_t = V_t(x',i) + \hat{k}_t, \end{aligned}$$

for all $i \in I$ and $t \in T$. We then prove Equation (3.24) by induction on t. Recall that $V_{N+1}(x,i) = V_{N+1}(x',i) = 0$ and $\hat{k}_{N+1} = 0$, and so

$$\begin{aligned} W_N(x,i) - W_N(x',i) &= G_N(x,i) - G_N(x',i) + \mathcal{E}V_{N+1}(x,i) - \mathcal{E}V_{N+1}(x',i) \\ &= G_N(x') - G_N(x) + \hat{k}_{N+1}. \end{aligned}$$

Now suppose that Equation (3.24) holds for all $t \geq l+1$; we need to show that it holds for $t = l$. Note that by the induction hypothesis, Lemma 3.3.1, and Equation (3.23), we have $x' \oplus q \leq x \oplus q$ and

$$\begin{aligned}
\mathcal{E}V_{t+1}(x,i) - \mathcal{E}V_{t+1}(x',i) &= E\left[V_{t+1}(x \oplus q_t, i_{t+1}) - V_{t+1}(x' \oplus q_t, i_{t+1}) | i_t = i\right] \\
&= \sum_{j=1}^{M} \sum_{q=0}^{Q} [V_{t+1}(x \oplus q, t) - V_{t+1}(x' \oplus q, t)][D_q]_{i,j} \\
&\leq \hat{k}_{t+1}.
\end{aligned}$$

Then we have

$$W_t(x,i) - W_t(x',i) \leq G_t(x,i) - G_t(x',i) + \hat{k}_{t+1}.$$

This completes our proof. □

The proof of Lemma 3.3.4 is purely analytic. As for the case of Lemma 3.3.2, we construct an alternative proof of (3.23) which may be made rigorous. If the initial system state at the beginning of period t before dispatch is (x,i). Suppose we clear everything at the beginning of period t and start from an empty state, and after that we mimic the optimal clearing policy as if the initial system state before dispatch is (x',i). Consequently due to Lemma 3.3.2, this policy incurs expected total cost not exceeding $V_t(x',i) + \hat{k}_t$. On the other hand, this policy just described cannot be better than the optimal policy, thus (3.23) must hold. A similar argument can be used to establish (3.24).

Lemma 3.3.4 gives an upper bound on the value function for different initial system states. Essentially, it means that for any initial system state (x,i), the expected total cost under the optimal policy must not exceed the expected total cost of clearing everything first, and then starting from state $(\emptyset,i)$ under the same optimal policy.

So far, we have shown the optimal clearing decisions based on the partial order of Φ. However, that partial order may not always apply. For instance, x and $x \oplus y$ do not always follow the partial order, for all $x, y \in \Phi$. The following lemma provides another decision

guideline for this scenario.

Proposition 3.3.5. *Under Condition 3.2.1*

(i) $G_t(x \oplus z, i) \leq G_t(x \oplus y \oplus z, i)$.

(ii) $V_t(x \oplus z, i) \leq V_t(x \oplus y \oplus z, i)$ *and* $0 \leq G_t(x \oplus y \oplus z, i) - G_t(x \oplus z, i) \leq W_t(x \oplus y \oplus z, i) - W_t(x \oplus z, i)$.

(iii) $V_t(x \oplus y \oplus z, i) \leq V_t(x \oplus z, i) + \hat{k}_t$ *and* $W_t(x \oplus y \oplus z, i) - W_t(x \oplus z, i) \leq G_t(x \oplus y \oplus z, i) - G_t(x \oplus z, i) + \hat{k}_{t+1}$.

for all $x, y, z \in \Phi$, $i \in I$, *and* $t \in T$.

Proof. We begin by proving the first half of part (i).

$$
\begin{aligned}
&G_t(x \oplus y \oplus z, i) - G_t(x \oplus z, i) \\
&= -\left[C_t(x \oplus y \oplus z) - C_t(x \oplus z)\right] + \left[H_t(x \oplus y \oplus z) - H_t(x \oplus z)\right] + \\
&\sum_{j=1}^{M} \sum_{q=0}^{Q} \left[C_{t+1}(x \oplus y \oplus z \oplus q) - C_{t+1}(x \oplus z \oplus q)\right] [D_q]_{i,j} \\
&= -C_t(y) + C_{t+1}(y) + H_t(x \oplus y \oplus z) - H_t(x \oplus z) \\
&\geq -C_t(y) + C_{t+1}(y) + H_t(x \oplus [0, \ldots, 0] \oplus z) + H_t(y \oplus [0, \ldots, 0]) - H_t(x \oplus z) \\
&\geq -C_t(y) + C_{t+1}(y) + H_t(y) + H_t(x \oplus [0, \ldots, 0] \oplus z) - H_t(x \oplus z) \geq 0.
\end{aligned}
$$

The first equality comes directly from the definition of function G_t, the second equality follows from part (ii) of Condition 3.2.1, and the inequalities result from parts (iv) and (v). Base on part (i), part (ii) follows the same induction proof as for Lemma 3.3.2, and part (iii) follows the same induction proof as that of Lemma 3.3.4. Details are omitted here. □

The next theorem guarantees that if G_t is non-increasing over time, then so is V_t. This is intuitive because we are accounting for costs over fewer periods of time as we approach the end of the planning horizon.

Theorem 3.3.6. *Under Condition 3.2.1, if* $G_t(x,i) \geq G_{t+1}(x,i)$*, then we have*

$$W_t(x,i) \geq W_{t+1}(x,i) \tag{3.25}$$

and

$$V_t(x,i) \geq V_{t+1}(x,i) \tag{3.26}$$

for all $x \in \Phi$*,* $i \in I$ *and* $t \in T$*.*

Proof. We prove Equations (3.25) and (3.26) simultaneously by induction. First, we have

$$\begin{aligned} W_{N-1}(x,i) - W_N(x,i) &= G_{N-1}(x,i) - G_N(x,i) + \mathcal{E}V_N(x,i) - \mathcal{E}V_{N+1}(x,i) \\ &\geq \mathcal{E}V_N(x,i) \geq 0, \end{aligned}$$

since $V_{N+1}(x,i) = 0$. This leads directly to $V_{N-1}(x,i) \geq V_N(x,i)$ because $W_{N-1}(\emptyset,i) \geq W_N(\emptyset,i)$ and $K_{N-1}(x) \geq K_N(x)$. Now suppose for all $t \geq l+1$, Equations (3.25) and (3.26) hold true. We have

$$\begin{aligned} W_t(x,i) - W_{t+1}(x,i) &= G_t(x,i) - G_{t+1}(x,i) + \mathcal{E}V_{t+1}(x,i) - \mathcal{E}V_{t+2}(x,i) \\ &= G_t(x,i) - G_{t+1}(x,i) + \mathcal{E}V_{t+1}(x,i) - \mathcal{E}V_{t+2}(x,i) \\ &\geq \sum_{j=1}^{M} \sum_{q=0}^{Q} [V_{t+1}(x \oplus q, i) - V_{t+2}(x \oplus q, i)][D_q]_{i,j} \geq 0. \end{aligned}$$

Finally, we have $V_t(x,i) \geq V_{t+1}(x,i)$, which completes our proof. □

3.3.2 Policy Boundaries

In some system states, the optimal clearing decisions are very obvious based on system parameters. For example, if the delay penalty cost for carrying the input for one more period exceeds the clearing costs, then it is obviously optimal to clear the system immediately. Therefore, if we can accurately identify those system states, we can prescribe the optimal

clearing decisions for them directly.

Let us define a pair of nonnegative real numbers $\underline{v}_{t,i} \leq \bar{v}_{t,i}$ as

$$\underline{v}_{t,i} = G_t(\emptyset, i) + \hat{k}_t - \hat{k}_{t+1}; \tag{3.27}$$

and

$$\bar{v}_{t,i} = G_t(\emptyset, i) + \hat{k}_t, \tag{3.28}$$

for all $i \in I$ and $t \in T$.

Lemma 3.3.7. *Under Condition 3.2.1, for any system state $(x, i) \in \Omega$, if $G_t(x, i) < \underline{v}_{t,i}$, then $W_t(\emptyset, i) + \hat{k}_t > W_t(x, i)$; on the other hand, if $G_t(x, i) > \bar{v}_{t,i}$, then $W_t(\emptyset, i) + \hat{k}_t < W_t(x, i)$.*

Proof. From Condition 3.2.1, we have $G_t(\emptyset, i) \leq G_t(x, i)$ for all $(x, i) \in \Omega$. The first inequality then follows from Lemma 3.3.4, such that

$$W_t(\emptyset, i) - W_t(x, i) + \hat{k}_t \geq G_t(\emptyset, i) - G_t(x, i) - \hat{k}_{t+1} + \hat{k}_t = \underline{v}_{t,i} - G_t(x, i) > 0,$$

whenever $G_t(x, i) < \underline{v}_{t,i}$. Similarly, by Lemma 3.3.2

$$W_t(\emptyset, i) - W_t(x, i) + \hat{k}_t \leq G_t(\emptyset, i) - G_t(x, i) + \hat{k}_t = \bar{v}_{t,i} - G_t(x, i) < 0,$$

whenever $G_t(x, i) > \bar{v}_{t,i}$. □

Lemma 3.3.7 provides two useful bounds for making better clearing decisions. The result can be interpreted as, for any given system state (x, i) at the beginning of period t, if $G_t(x, i) < \underline{v}_{t,i}$, it is optimal to continue to accumulate; on the other hand, if $G_t(x, i) > \bar{v}_{t,i}$, then it is optimal to clear all accumulated quantities. The use of these bounds can be further simplified by the following proposition.

Proposition 3.3.8. *Suppose that Condition 3.2.1 holds. For any system state $(x, i) \in \Omega$, if $H_t(x) - C_t(x) + C_{t+1}(x) < \hat{k}_t - \hat{k}_{t+1}$, then the optimal action is not to clear x; on the other hand, if $H_t(x) - C_t(x) + C_{t+1}(x) > \hat{k}_t$, then the optimal action is to clear x.*

Proof. We only need to observe that because of parts (ii) and (iii) of Condition 3.2.1 and the fact that $C_t(\emptyset) = H_t(\emptyset) = 0$,

$$G_t(x, i) - G_t(\emptyset, i) = H_t(x) - C_t(x) + C_{t+1}(x).$$

The rest follows directly from Lemma 3.3.7. □

Lemma 3.3.9. *Under Condition 3.2.1, for every $i \in I$ and $t \in T$, there are only finitely many sequences which do not require clearing under the optimal policy.*

Proof. The proof follows directly from Condition 3.2.1.(vi) and Proposition 3.3.8, because there are only a finite number of sequences $x \in \Phi$ for which $H_t(x) - C_t(x) + C_{t+1}(x) \leq \hat{k}_t$. □

Lemma 3.3.9 guarantees clearing if the input sequence gets large enough with respect to the partial order. Clearing will thus occur if the input sizes are great enough, or the input ages are old enough.

Recall from Condition 2.2.1, a clearing policy is said to be *logical* if (i) it never clears an empty system, (ii) clearing system content x implies it must clear any other system with contents $x \oplus y$, and (iii) clearing system content x implies it must clear any other system of contents y such that $x \leq y$. Also recall from Condition 2.2.2, a policy is feasible if it guarantees clearing after finite accumulation and finite time. The following theorem summarizes the characteristics of an MD optimal policy.

Theorem 3.3.10. *Under Condition 3.2.1, an MD optimal policy is always logical and feasible, i.e., it satisfies Condition 2.2.1 and 2.2.2, and for each period $t \in T$, the set $\Psi_{(r_t^*)}$ has a tree structure and finite size.*

Proof. Condition 2.2.1.(i) is trivial to verify but necessary. Condition 2.2.1.(ii) is a direct result from Proposition 3.3.5 since $x \oplus \emptyset = x$ and $x \oplus \emptyset \oplus y = x \oplus y$. More specifically, for any system state (x, i) at the beginning of period t, if the optimal action is to clear, then $x \neq \emptyset$ and $W_t(\emptyset, i) + \hat{k}_t < W_t(x, i)$. At the same time, $W_t(x, i) \leq W_t(x \oplus y, i)$ and $K_t(x \oplus y) = \hat{k}_t$ because $x \oplus y \neq \emptyset$. Together, we have $W_t(\emptyset, i) + \hat{k}_t < W_t(x \oplus y, i)$. Similarly, Condition 2.2.1.(iii) follows from Lemma 3.3.2. Lemma 3.3.9 guarantees the feasibility of the optimal policy. The finite tree structure of $\Psi_{(r_t^*)}$ is implied by Properties 2.2.1 and 2.2.2. □

3.4 State-Dependent Threshold Policies

In the previous section, some features of the optimal policy were found, but the actual representation of the optimal policy was still very cumbersome. In this section, we show that, subject to some special conditions on the delay penalty function, the optimal policy can actually be a *state-dependent threshold policy*. That policy is parameterized by a single number, as opposed to the sets $\Lambda_{r_t^*}$ for each period $t \in T$.

3.4.1 Special Delay Penalty Functions

We first introduce a class of delay penalty functions that depends entirely on the cumulative quantities. This type of function is consistent with the "holding cost function" in the existing literature of stochastic clearing systems.

Definition 3.4.1. *For any sequence $y = [y_{[l]}, \ldots, y_{[1]}]$ and $t \in T$, we say a delay penalty function H_t is "purely quantity-dependent", if*

$$H_t(y) = H_t^{\mathrm{Q}}(|y|), \tag{3.29}$$

where H_t^{Q} is a non-decreasing function with respect to the cumulative quantities $|y|$.

Next, for any sequence $y = [y_{[l]}, \ldots, y_{[1]}]$, we can make another binary sequence, denoted by y^{A}, to capture the ages of individual outstanding inputs, such that

$$y^{\mathrm{A}} = [\delta_{\{y_{[l]}>0\}},\ \ldots,\ \delta_{\{y_{[1]}>0\}}]. \tag{3.30}$$

Similar to Definition 3.4.1, we can introduce another type of delay penalty function that depends only on the input ages.

Definition 3.4.2. *For any sequence $y = [y_{[l]}, \ldots, y_{[1]}]$ and $t \in T$, we say a delay penalty function H_t is "purely age-dependent", if*

$$H_t(y) = H_t^{\mathrm{A}}(y^{\mathrm{A}}), \tag{3.31}$$

where H_t^{A} is a function with respect to individual input ages, and

$$H_t^{\mathrm{A}}(x^{\mathrm{A}}) \leq H_t^{\mathrm{A}}(y^{\mathrm{A}}) \tag{3.32}$$

if and only if

$$\sum_{j=1}^{l} j x_{[j]}^{\mathrm{A}} \leq \sum_{j=1}^{l} j y_{[j]}^{\mathrm{A}}, \tag{3.33}$$

where Equation (3.33) means that the total age in x is no greater than the total age in y.

Now, we observe that purely quantity-dependent or purely age-dependent delay penalty functions have a special property.

Proposition 3.4.1. *For any purely quantity-dependent or purely age-dependent delay penalty function, $H_t(x) \leq H_t(y)$ implies $H_{t+1}(x \oplus q) \leq H_{t+1}(y \oplus q)$, for all $t \in T$, $x, y \in \Phi$, and $0 \leq q \leq Q$.*

Proof. For purely quantity-dependent penalty, the proof is straightforward, since $|y| \geq |x|$

implies $|y \oplus q| \geq |x \oplus q|$. For purely age-dependent penalty, $H_t(x) \leq H_t(y)$ suggests that

$$\sum_{j=1}^{l} j x_{[j]}^{\mathrm{A}} \leq \sum_{j=1}^{l} j y_{[j]}^{\mathrm{A}}.$$

We must then have

$$\sum_{j=1}^{l} (j+1) x_{[j]}^{\mathrm{A}} + \delta_{\{q>0\}} \leq \sum_{j=1}^{l} (j+1) y_{[j]}^{\mathrm{A}} + \delta_{\{q>0\}},$$

which then leads to $H_{t+1}(x \oplus q) \leq H_{t+1}(y \oplus q)$. □

For the remainder of this section, we focus on cost functions satisfying Condition 3.4.1.

Condition 3.4.1.

(i) The delay penalty function is either purely quantity-dependent or purely age-dependent.

(ii) $C_{t+1}(x) - C_t(x) = 0$ *for all* $x \in \Phi$ *and* $t \in T$*, i.e., the variable clearing cost rate is constant over the planning horizon.*

Proposition 3.4.2. *Under Conditions 3.2.1 and 3.4.1, if* $H_t(x) \leq H_t(y)$ *for some* $x \in \Phi$ *and* $t \in T$*, then we must have, for all* $i, j \in I$ *and* $q = 0, 1, \ldots, Q$*:*

(i) $G_t(x, i) \leq G_t(y, i)$ *and* $G_{t+1}(x \oplus q, j) \leq G_{t+1}(y \oplus q, j)$.

(ii) $V_t(x, i) \leq V_t(y, i)$ *and* $0 \leq G_t(y, i) - G_t(x, i) \leq W_t(y, i) - W_t(x, i)$.

(iii) $V_t(y, i) \leq V_t(x, i) + \hat{k}_t$ *and* $W_t(y, i) - W_t(x, i) \leq G_t(y, i) - G_t(x, i) + \hat{k}_{t+1}$.

Proof. According to Condition 3.4.1.(ii), Condition 3.2.1.(ii), and Equation (3.10), we have

$$\begin{aligned}
0 &\leq H_t(y) - H_t(x) \\
&= (H_t(y) - H_t(x)) + (C_{t+1}(y) - C_t(y)) - (C_{t+1}(x) - C_t(x)) \\
&= \left(H_t(y) - C_t(y) + \sum_{q=0}^{Q} C_{t+1}(y \oplus q) f_i(q) \right) - \left(H_t(x) - C_t(x) + \sum_{q=0}^{Q} C_{t+1}(x \oplus q) f_i(q) \right) \\
&= G_t(y, i) - G_t(x, i).
\end{aligned}$$

Similarly, since H_t is either purely quantity-dependent or purely age-dependent, Proposition 3.4.1 implies

$$0 \leq H_{t+1}(y \oplus q) - H_{t+1}(x \oplus q) = G_{t+1}(y \oplus q, j) - G_{t+1}(x \oplus q, j),$$

which completes the proof of part (i). Parts (ii) and (iii) follow the same induction proofs as those of Lemmas 3.3.2 and 3.3.4. Details are omitted here. □

Proposition 3.4.2 is very useful because, instead of having to check the actual system state against the sets $\Lambda_{r_t^*}$, it allows us to make optimal clearing decisions based on H_t. This leads to a state-dependent threshold policy which can be optimal. We shall define this type of threshold policy in the next section.

3.4.2 Characteristics of State-Dependent Threshold Policies

Definition 3.4.3. *A state-dependent threshold clearing policy in a finite horizon is denoted as* $\pi^\tau = (r_1^\tau, \ldots, r_N^\tau)$. *For period* t, *the clearing rule* r_t^τ *is determined by a set of parameters* $\{\tau_{t,1}, \ldots, \tau_{t,M}\}$, *for which we have*

$$r_t^\tau(x, i) = \begin{cases} 0, & \text{if } H_t(x) \leq \tau_{t,i} \\ 1, & \text{otherwise} \end{cases}, \tag{3.34}$$

for all $(x, i) \in \Omega$.

The following theorem shows that under special conditions, the optimal clearing policy can be a state-dependent threshold policy.

Theorem 3.4.3. *If Conditions 3.2.1 and 3.4.1 are satisfied, then the optimal clearing policy can be a state-dependent threshold clearing policy denoted as* π^{τ^*}. *More specifically, for each underlying state* $i \in I$, *there exists a non-empty sequence* $x_t^*(i)$ *such that*

$$x_t^*(i) = \arg\max_{x \in \Phi} \{H_t(x) : W_t(x, i) \leq W_t(\emptyset, i) + K_t(x)\}, \tag{3.35}$$

and

$$\tau_t^*(i) = H_t(x_t^*(i)), \tag{3.36}$$

for all $i \in I$ *and* $t \in T$.

Proof. First, it is easy to see that Equation (3.35) implies that $x_t^*(i)$ is the sequence with the highest delay penalty H_t among all sequences which are better not to be cleared in period t, for underlying state i. The existence of such $x_t^*(i)$ is guaranteed by Lemma 3.3.9. Then in any period $t \in T$ and underlying state i, for any non-empty system content y such that $H_t(y) > H_t(x_t^*(i))$, Proposition 3.4.2 suggests that $G_t(y, i) > G_t(x_t^*(i), i)$ and $W_t(y, i) > W_t(x_t^*(i), i)$. Consequently, Equation (3.35) shows that

$$W_t(y, i) > W_t(\emptyset, i) + K_t(y),$$

which means clearing is preferred. On the other hand, for any non-empty system state (z, i) such that $H_t(z) \leq H_t(x_t^*(i))$, we have $G_t(z, i) \leq G_t(x_t^*(i), i)$ and $W_t(z, i) \leq W_t(x_t^*(i), i)$, this leads to

$$W_t(z, i) \leq W_t(\emptyset, i) + K_t(z),$$

which indicates that clearing is not necessary. □

Theorem 3.4.3 confirms a well-known result in inventory ([70], [82], and [90]) and shipment consolidation ([12], [13], and [16]) problems, which states that if the delay penalty depends only on the cumulative quantities, the optimal policy can be a quantity threshold policy. Details are given in the following corollary.

Corollary 3.4.4. *If Condition 3.2.1 is satisfied and the delay penalty function is purely quantity-dependent, then the optimal clearing policy can be a state-dependent quantity threshold clearing policy denoted as π^{Q^*}. More specifically, there exists a non-empty sequence $x_t^*(i)$ such that*

$$x_t^*(i) = \arg\max_{x\in\Phi}\{|x| : W_t(x,i) \leq W_t(\emptyset,i) + K_t(x)\}, \tag{3.37}$$

and

$$Q_t^*(i) = |x_t^*(i)|, \tag{3.38}$$

where $|x|$ is the sequence sum of x defined in Definition 2.1.1, for all $i \in I$ and $t \in T$.

Proof. For purely quantity-dependent delay penalty functions, H_t is a function of the cumulative quantities, thus, G_t is also a function of the cumulative quantities. Therefore, $H_t(y) > H_t(x_t^*(i))$ if and only if $|y| > |x_t^*(i)|$, which completes our proof. □

3.5 Computing the Optimal Policy Parameters

For discrete-time and discrete-quantity stochastic clearing systems over a finite horizon, Theorem 3.3.10 states that $\Psi_{(r_t^*)}$ has a finite tree structure. Therefore, in every decision epoch t, we can assume that only a finite subset of system states is reached. More specifically, Proposition 3.3.8 suggests that if $x_t = x$ such that $H_t(x) - C_t(x) + C_{t+1}(x) > \hat{k}_t$, then it is optimal to clear the system immediately, i.e., add $K_t(x) + C_t(x)$ to the expected total cost and assume that the system continues from empty state $(\emptyset, i)$.

The following backward induction algorithm utilizes the tree structure to help construct the optimal policy.

Algorithm II: Finite Horizon Optimal Policy Algorithm

II.1 If $H_N(x) - C_N(x) + C_{N+1}(x) > \hat{k}_N$, let $r_N(x,i) = 1$; otherwise, let $r_N(x,i) = 0$. Use BFT Procedure I to create $\Psi_{(r_N)}$.

II.2 For each $(x,i) \in \Psi_{(r_N)} \times I$, set $V^{N+1}(x,i) = 0$. Initialize counter $n = N$.

II.3 If $H_{n-1}(x) - C_{n-1}(x) + C_n(x) > \hat{k}_{n-1}$, let $r_{n-1}(x,i) = 1$; otherwise, let $r_{n-1}(x,i) = 0$. Use BFT procedure I to create $\Psi_{(r_{n-1})}$.

II.4 For each $x \in \Psi_{(r_{n-1})}$, $j \in I$, and $q = 0, 1, \ldots, Q$, create $y = x \oplus q$ and look up $V^{n+1}(y,j)$. If not found, set $V^{n+1}(y,j) = K_{n+1}(y) + C_{n+1}(y) + V^{n+1}(\emptyset, j)$.

II.5 Compute $W_n(x,i)$ and $W_n(\emptyset,i)$ by Equations (3.12) and (3.17). If $W_n(x,i) > W_n(\emptyset,i) + K_n(x)$, record $V_n(x,i) = W_n(\emptyset,i) + K_n(x)$ and assign $r_n^*(x,i) = 1$; otherwise, record $V_n(x,i) = W_n(x,i)$ and assign $r_n^*(x,i) = 0$. Set $n := n - 1$.

II.6 If $n = 0$, stop and report $r_t^*(x,i)$ and $V_t(x,i)$, for all $(x,i) \in \Lambda_{r_t^*}$ and $t \in T$; otherwise, go back to Step 3.

If Conditions 3.2.1 and 3.4.1 are both satisfied, we can output $\tau_t^*(i)$ instead of r_t^* for every $i \in I$ and $t \in T$.

Recall from Section 2.1.4, we can denote the oldest input age allowed by the optimal policy in period t as $\bar{L}_t^*$. This age limit is very important because $|\Psi_{(r_t^*)}| \leq Q^{\bar{L}_t^*}$, but without knowing the elements in $\Psi_{(r_t^*)}$, we can hardly compute $\bar{L}_t^*$. Here we present a way to estimate $\bar{L}_t^*$.

We see that

$$\bar{L}_t^* \leq \max_{x\in\Phi}\{\mathcal{L}(x) : H_t(x) - C_t(x) + C_{t+1}(x) \leq \hat{k}_t\}, \tag{3.39}$$

for all $t \in T$. Therefore, if we assume integer input quantities and $C_{t+1}(x) - C_t(x)$ is negligible, we can estimate $\bar{L}_t^*$ by a single number

$$\tilde{L} = \max_{t\in T}\{\max\{l_t + 1 : H_t(1 \oplus 0_{l_t}) \leq \hat{k}_t\}\}, \tag{3.40}$$

where 0_{l_t} is a sequence of zeros of length l_t. The justification for such estimation follows from Proposition 3.3.8 and the fact that $1 \oplus 0_{l_t}$ is the longest possible integer sequence for the delay penalty to not exceed the fixed clearing cost in period t. This estimate can be used to analyze the complexities of Algorithm I.

Time Complexity: The time complexity of BFT Procedure I has an order of growth of $\mathcal{O}(MQ^{\tilde{L}})$. The bulk of the work of the algorithm is in Step III.4, where we need to look up $V_{n+1}(x \oplus q, j)$ for all $x \in \Psi_{(r_n^*)}$, $j \in I$ and $q = 0, 1, \ldots, Q$. Therefore, the time complexity of the algorithm is $\mathcal{O}(M^2NQ^{\tilde{L}})$ over N periods.

Memory Complexity: Since each sequence in $\Psi_{(r_t^*)}$ must be stored for all $t \in T$, and any other information stored requires significantly less space, the memory requirement is $\mathcal{O}(\tilde{L}NQ^{\tilde{L}})$.

Please refer to Appendix C for a numerical example to verify the complexity estimates. We demonstrate Algorithm II through the Example 3.5.1.

Example 3.5.1. *Consider a stochastic clearing system with the following parameters.*

(i) The planning horizon has three periods.

(ii) The input process is given as the BMAP in Example 2.1.1.(ii).

(iii) The delay penalty cost is given as in Example 2.3.1.(d).

(iv) The fixed clearing cost is $K_t(w_t) = (4 - t) \cdot \delta_{\{|w_t|>0\}}$.

(v) The variable clearing cost is $C_t(w_t) = 0.5(4-t)|w_t|$.

The optimal clearing policy and the corresponding value functions are summarized in Tables 3.2 to 3.4.

Result Summary	$r_3^*(x,1)$	$r_3^*(x,2)$	$V_3(x,1)$	$V_3(x,2)$
$\emptyset$	0	0	0.5468	0.1823
$x=[1]$	0	0	1.0733	0.7088
$x=[2]$	1	1	2.1668	1.8023
$x=[3]$	1	1	2.1668	1.8023
$x=[1,0]$	0	0	1.5593	1.1948
$x=[1,1]$	0	0	2.0858	1.7213
$x=[1,2]$	1	1	2.5718	2.2073
$x=[2,0]$	1	1	2.1668	1.8023
$x=[2,1]$	1	1	2.5718	2.2073
$x=[1,0,0]$	1	1	1.7618	1.3973
$x=[1,0,1]$	1	1	2.1668	1.8023
$x=[1,0,2]$	1	1	2.5718	2.2073
$x=[1,1,0]$	1	1	2.1668	1.8023
$x=[1,1,1]$	1	1	2.5718	2.2073
$x=[1,1,2]$	1	1	2.9768	2.6123
$x=[1,0,0,0]$	1	1	1.7618	1.3973
$x=[1,0,0,1]$	1	1	2.1668	1.8023

Table 3.2: Summary of Optimal Clearing Rule in Period 3 for Example 3.5.1

The cost functions given in Example 3.5.1 clearly satisfy Condition 3.2.1. The numerical results of Example 3.5.1 show that the optimal clearing rules depend on the period t and the underlying state i, and demonstrate the characteristics proven in Section 3.3.

In the next example, we show a stochastic clearing problem with an optimal state-dependent threshold clearing policy.

Example 3.5.2. *Consider a stochastic clearing system with the following parameters.*

(i) The planning horizon has three periods.

(ii) The input process is given as the BMAP in Example 2.1.1.(ii).

x	$r_2^*(x,1)$	$r_2^*(x,2)$	$V_2(x,1)$	$V_2(x,2)$
$\emptyset$	0	0	1.6403	0.5387
$x=[1]$	0	0	2.4683	1.6704
$x=[2]$	1	1	4.3403	3.2387
$x=[1,0]$	0	0	3.0528	2.3562
$x=[1,1]$	0	0	3.6378	2.9412
$x=[1,2]$	1	1	4.7903	3.6887
$x=[2,0]$	1	1	4.3403	3.2387
$x=[2,1]$	1	1	4.7903	3.6887
$x=[1,0,0]$	1	1	3.8903	2.7887
$x=[1,0,1]$	1	1	4.3403	3.2387
$x=[1,0,2]$	1	1	4.7903	3.6887
$x=[1,1,0]$	1	1	4.3403	3.2387
$x=[1,1,1]$	1	1	4.7903	3.6887
$x=[1,1,2]$	1	1	5.2403	4.1387
$x=[1,0,0,0]$	1	1	3.8903	2.7887
$x=[1,0,0,1]$	1	1	4.3403	3.2387

Table 3.3: Summary of Optimal Clearing Rule in Period 2 for Example 3.5.1

x	$r_1^*(x,1)$	$r_1^*(x,2)$	$V_1(x,1)$	$V_1(x,2)$
$\emptyset$	0	0	3.4076	1.2492
$x=[1]$	0	0	4.3768	2.9387
$x=[2]$	1	1	7.4076	5.2492
$x=[1,0]$	0	0	5.2551	3.9238
$x=[1,1]$	0	0	5.9051	4.5738
$x=[1,2]$	1	1	7.9076	5.7492
$x=[2,0]$	1	1	7.4076	5.2492
$x=[2,1]$	1	1	7.9076	5.7492
$x=[1,0,0]$	1	1	6.9076	4.7492]
$x=[1,0,1]$	1	1	7.4076	5.2492
$x=[1,0,2]$	1	1	7.9076	5.7492
$x=[1,1,0]$	1	1	7.4076	5.2492
$x=[1,1,1]$	1	1	7.9076	5.7492
$x=[1,1,2]$	1	1	8.4076	6.2492
$x=[1,0,0,0]$	1	1	6.9076	4.7492
$x=[1,0,0,1]$	1	1	7.4076	5.2492

Table 3.4: Summary of Optimal Clearing Rule in Period 1 for Example 3.5.1

(iii) The delay penalty cost is given as in Example 2.3.1.(c).

(iv) The fixed clearing cost is $K_t(w_t) = (4-t) \cdot \delta_{\{|w_t|>0\}}$.

(v) The variable clearing cost is $C_t(w_t) = 0.5|w_t|$.

The optimal clearing policy and the corresponding value functions are summarized in Tables 3.5 to 3.7.

x	$r_3^*(x,1)$	$r_3^*(x,2)$	$V_3(x,1)$	$V_3(x,2)$	$H(x,2)$
$\emptyset$	0	0	0.5468	0.1823	0
[1]	0	0	1.3163	0.9518	0.5
[2]	0	0	1.6808	1.3163	0.5
[1,0]	1	1	1.7618	1.3973	2
[1,1]	1	1	2.1668	1.8023	2.5
[1,2]	1	1	2.5718	2.2073	2.5
[2,0]	1	1	2.1668	1.8023	2
[2,1]	1	1	2.5718	2.2073	2.5
[2,2]	1	1	2.9768	2.6123	2.5

Table 3.5: Summary of Optimal Clearing Rule in Period 3 for Example 3.5.2

x	$r_2^*(x,1)$	$r_2^*(x,2)$	$V_2(x,1)$	$V_2(x,2)$	$H(x,2)$
$\emptyset$	0	0	1.4216	0.5630	0
[1]	0	0	2.7828	2.0862	0.5
[2]	0	0	3.1878	2.4912	0.5
[1,0]	1	1	3.6716	2.8130	2
[1,1]	1	1	4.1216	3.2630	2.5
[1,2]	1	1	4.5716	3.7130	2.5
[2,0]	1	1	4.1216	3.2630	2
[2,1]	1	1	4.5716	3.7130	2.5
[2,2]	1	1	5.0216	4.1630	2.5

Table 3.6: Summary of Optimal Clearing Rule in Period 2 for Example 3.5.2

The cost functions given in Example 3.5.2 clearly satisfy both Conditions 3.2.1 and 3.4.1. The numerical results of Example 3.5.2 show that the optimal clearing rules can be converted into state-dependent threshold rules defined in Section 3.4. In the preceding example, it is found that the optimal threshold policy parameter is $\tau_{t,i}^* = 0.5$, for $1 \leq t \leq 3$ and $i = 1, 2, \ldots$.

x	$r_1^*(x,1)$	$r_1^*(x,2)$	$V_1(x,1)$	$V_1(x,2)$	$H(x,2)$
$\emptyset$	0	0	2.8184	1.2921	0
[1]	0	0	4.7607	3.6238	0.5
[2]	0	0	5.2107	4.0738	0.5
[1,0]	1	1	6.3184	4.7921	2
[1,1]	1	1	6.8184	5.2921	2.5
[1,2]	1	1	7.3184	5.7921	2.5
[2,0]	1	1	6.8184	5.2921	2
[2,1]	1	1	7.3184	5.7921	2.5
[2,2]	1	1	7.8184	6.2921	2.5

Table 3.7: Summary of Optimal Clearing Rule in Period 1 for Example 3.5.2

3.6 Summary of Results

In conclusion, for the expected total cost problem over a finite horizon, we obtain the following results in this chapter. First, we construct an algorithm to evaluate any given clearing policy. Then, we show that an order of the optimal total cost over finite horizon is induced by a partial order of the initial system state space. Using this order, we prove that an optimal clearing policy exists and such optimal policy never allows for partial clearing, i.e., if it is optimal to clear, we shall clear everything. The order of the total cost function also gives us some useful bounds on the optimal clearing policy parameters.

Next, we demonstrate that if the cost structure satisfies some additional conditions, then the optimal clearing policy can be a threshold policy which is parameterized by simple threshold values. These threshold policies are much easier to compute, record, and deploy, plus they can be calculated for relatively larger problem instances which we otherwise would not be able to handle.

Lastly, we construct a backward induction algorithm to compute the optimal clearing policy parameters for our model. Through some numerical examples, we show that our analytical results hold and the optimal policy parameters do indeed have a tree structure. Complexity analyses of the optimization algorithm reveal that the time and memory complexities both grow at an exponential rate, but our model can still solve some real life

problems of moderate size.

After studying the problem over a finite horizon, one may naturally ask if the results in this chapter can be carried over to the problem over an *infinite* horizon. The standard approach to answer this question is by allowing the planning horizon to be of length T, and taking the limit as $T \to \infty$ of the formulas for the finite horizon problem. In the next chapter, we will give this approach a try.

Chapter 4

Expected Total Discounted Cost Model over Infinite Horizon

In this chapter, we study the stochastic clearing problems over an *infinite* planning horizon, with the expected total discounted cost as our objective function. We first present an algorithm to compute the expected total cost for any given clearing policy. Then we extend the results of Section 3.3 to show some characteristics of the optimal policy. The state-dependent threshold policy introduced in Section 3.4 is proven to be optimal for special types of delay penalty functions. Lastly, we present three different methods to compute the optimal policy parameters. The modelling assumptions and notations used in this chapter are summarized below.

(i) The planning horizon is infinite, i.e., $T = 1, 2, \ldots$.

(ii) The model is in discrete time and discrete quantities, and the input process is modelled as a *BMAP*.

(iii) The clearing policy is given by a stationary policy $\pi = (r, r, \ldots)$; thus, π and r can be used interchangeably.

(iv) The pre-clearing system state process $\{(x_t, i_t), t = 1, 2, \ldots\}$ is the Markov chain of interest here.

(v) The cost functions are given by H, K, and C, which are stationary over time. (We can add the subscript t, if necessary, and extend the procedures below).

(vi) The costs incurred in period t are discounted by α^{t-1}, where $0 \leq \alpha < 1$ is the *discount factor*.

4.1 Policy Evaluation

Analogous to Section 3.1, the expected total discounted cost over the infinite planning horizon can be found as

$$\begin{aligned} \mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>}(x,i) &= E\left[\sum_{t=1}^{\infty} \alpha^{t-1} \left(H(y_t) + K(w_t) + C(w_t)\right) \middle| (x_1, i_1) = (x,i)\right] \\ &= E\left[\sum_{t=1}^{\infty} \alpha^{t-1} u^r(x_t, i_t) \middle| (x_1, i_1) = (x,i)\right] \\ &= u^r(x,i) + \alpha \sum_{(x',j)\in\Lambda_{(r)}} P^{(x,r)}_{(x,i),(x',j)} \mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>}(x',j), \end{aligned} \tag{4.1}$$

where $a_t = r(x_t, i_t)$, $P^{(x,r)}_{(x,i),(x',j)}$ are given by Equations (2.25) to (2.28), and $u^r(x,i)$ as in Equation (3.1).

Using the vector and matrix notation defined in Equations (2.29) and (2.30) to collect the underlying states together and expand $\Lambda_{(r)}$ to $\Psi_{(r)} \times I$, we can rewrite Equation (4.1) as

$$\mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>}(\boldsymbol{x}) = u^r(\boldsymbol{x}) + \alpha \sum_{q=0}^{Q} A^{(x,r)}_{x,x\oplus q} \mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>}(\boldsymbol{x} \oplus \boldsymbol{q}) + B^{(x,r)}_{x,q} \mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>}(\boldsymbol{q}), \tag{4.2}$$

where

$$\mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>}(\boldsymbol{x}) = \begin{pmatrix} \mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>}(x,1) \\ \vdots \\ \mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>}(x,M) \end{pmatrix}, u^r(\boldsymbol{x}) = \begin{pmatrix} u^r(x,1) \\ \vdots \\ u^r(x,M) \end{pmatrix}, \tag{4.3}$$

because we know that, for all $x \in \Psi_{(r)}$ and $q = 0, 1, \ldots, Q$, matrices $A^{(x,r)}_{x,x\oplus q}$ and $B^{(x,r)}_{x,q}$ are the only non-zero blocks of the transition probability matrix $P^{(x,r)}$ for $\{(x_t, i_t), t = 1, 2, \ldots\}$.

If we generalize this further by collecting all sequences in $\Psi_{(r)}$, we can rewrite Equation (4.2) as

$$\mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>} = u^r + \alpha P^{(x,r)} \mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>}. \tag{4.4}$$

Rearranging the terms, we get

$$(I - \alpha P^{(x,r)}) \mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>} = u^r. \tag{4.5}$$

Proposition 4.1.1. *For any stationary policy π, a unique solution exists for the system of equations in (4.5).*

Proof. From Equations (2.29) to (2.31), we see that

$$\sum_{j=1}^{M} \sum_{q=0}^{Q} \left(a^{(x,r)}_{(x,i),(x\oplus q,j)} + b^{(x,r)}_{(x,i),(q,j)} \right) = \sum_{j=1}^{M} D_{i,j} = 1, \forall (x, i) \in \Lambda_{(r)}.$$

In other words, the row sum of matrix $P^{(x,r)}$ is one. Using the standard definition of matrix norm, we know that $||P^{(x,r)}|| = 1$. Since $\alpha < 1$, the *spectral radius* of $\alpha P^{(x,r)}$, denoted by $\sigma(\alpha P^{(x,r)})$, is less than one because $\sigma(\alpha P^{(x,r)}) \leq ||\alpha P^{(x,r)}|| < 1$. Therefore, matrix-inverse theory indicates that $(I - \alpha P^{(x,r)})^{-1}$ exists and satisfies

$$(I - \alpha P^{(x,r)})^{-1} = \lim_{N\to\infty} \sum_{n=0}^{N} (\alpha P^{(x,r)})^n.$$

We can thus solve Equation (4.5) as

$$\mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>} = (I - \alpha P^{(x,r)})^{-1} u^r.$$

This completes our proof. □

We can compute the expected total discounted cost over the infinite horizon by Algorithm III.

Algorithm III: Infinite Horizon Policy Evaluation Algorithm

III.1 Construct $\Psi_{(r)}$ using BFT Procedure I.

III.2 For each $x \in \Psi_{(r)}$ and $q = 0, 1, \ldots, Q$, compute $A^{(x,r)}_{x,x\oplus q}$ and $B^{(x,r)}_{x,q}$ by Equations (2.29) and (2.30), and form the transition probability matrix $P^{(x,r)}$; also compute $u^r(\boldsymbol{x})$ by Equations (3.1) and (4.3), and form the vector u^r.

III.3 Solve Equation (4.5) by computing the inverse matrix $(I-\alpha P^{(x,r)})^{-1}$, then output $\mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>}(x,i)$.

Time Complexity: BFT Procedure I has time complexity $\mathcal{O}(MQ^{\bar{L}_{(r)}})$. The computation time in Step III.2 scales by $\mathcal{O}(MQ^{\bar{L}_{(r)}})$ because we only need to compute $r(x\oplus q, j)$ for each x, q and j to form block matrices A and B. Finally, Step III.3 requires solving an inverse matrix of size $MQ^{\bar{L}_{(r)}} \times MQ^{\bar{L}_{(r)}}$. Adopting a conservative method such as Gauss-Jordan elimination would yield a time complexity of $\mathcal{O}(M^3Q^{\bar{L}_{(r)}})$. Overall, the time complexity of Algorithm III is $\mathcal{O}(M^3Q^{\bar{L}_{(r)}})$.

Memory Complexity: The most memory is required to store the matrix $P^{(x,r)}$, which has a size bounded by $MQ^{\bar{L}_{(r)}} \times MQ^{\bar{L}_{(r)}}$, therefore, the memory requirement complexity is $\mathcal{O}(M^2Q^{\bar{L}_{(r)}})$.

Please refer to Appendix C for a numerical example to verify the complexity estimates. We demonstrate Algorithm III through Example 4.1.1.

Example 4.1.1. *Consider a stochastic clearing problem with the following parameters.*

(i) The discount factor $\alpha = 0.95$.

(ii) The input process is described in Example 2.1.1.(iii).

(iii) The delay penalty cost functions H_t is described in Example 2.3.1.(d).

(iv) The fixed clearing cost is given as $K(w_t) = 10 \cdot \delta_{\{|w_t|>0\}}$.

(v) The variable clearing cost is given as $C(w_t) = 0.5|w_t|$.

(vi) The clearing policy is given as

$$r(x_t, i_t) = \begin{cases} 0, & \text{if } |x_t| < 2 \text{ and } \mathcal{L}(x_t) < 3 \\ 1, & \text{otherwise} \end{cases}.$$

The expected total discounted costs for the different problems are computed by Algorithm III, and summarized in Table 4.1.

$\mathcal{C}^{\pi,\mathrm{T},\alpha}_{<1,\infty>}(x,i)$	$i=1$	$i=2$
$\emptyset$	3.4333	37.0888
$x=[1]$	7.6533	38.3517
$x=[2]$	8.4334	42.0888
$x=[3]$	8.9334	42.5888
$x=[1,0]$	7.9334	38.9517
$x=[1,1]$	8.4334	42.0888
$x=[1,2]$	8.9334	42.5888
$x=[1,3]$	9.4334	43.0888
$x=[2,0]$	8.4334	42.0888
$x=[2,1]$	8.9334	42.5888
$x=[1,0,0]$	7.9334	41.5888
$x=[1,0,1]$	8.4334	42.0888
$x=[1,0,2]$	8.9334	42.5888
$x=[1,0,3]$	9.4334	43.0888
$x=[1,1,0]$	8.4334	42.0888
$x=[1,1,1]$	8.9334	42.5888
$x=[1,1,2]$	9.4334	43.08883
$x=[1,0,0,0]$	7.9334	41.5888
$x=[1,0,0,1]$	8.4334	42.0888

Table 4.1: Summary of Expected Total Discounted Costs for Example 4.1.1

From the preceding numerical example, we can make the following observations. First, we notice that the input process is a Compound Markovian Input Process, and the input arrival rate in underlying state 1 is much larger than the rate in state 2. Therefore, we can see that for every sequence x in Table 3.1, $\mathcal{C}^{\pi,\mathrm{T},\alpha}_{<1,\infty>}(x,1) < \mathcal{C}^{\pi,\mathrm{T},\alpha}_{<1,\infty>}(x,2)$, and the difference is significant. Thus, the initial underlying state affects the expected total discounted cost.

Next, for $x = [1,0,1]$ and $x = [1,1,0]$, since dispatch is required immediately for both cases, the difference in the input sequences no longer matters as the total accumulated quantities are the same. Therefore, $\mathcal{C}^{\pi,\mathrm{T},\alpha}_{<1,\infty>}([1,0,1],i) = \mathcal{C}^{\pi,\mathrm{T},\alpha}_{<1,\infty>}([1,1,0],i)$, for each $i = 1,2$. These observations generally hold true and they can be used to characterize the optimal clearing policy.

4.2 Existence of Optimal Policies

To find the optimal clearing policy, we first need to establish the optimality equations for the infinite horizon problem. Recall from Section 4.1, we assumed that the optimal clearing policy is a stationary policy of the form $\pi^* = (r^*, r^*, \ldots)$, and the corresponding expected total discounted cost, denoted by $\mathcal{C}^{r^*,\mathrm{T},\alpha}_{<1,\infty>}(x,i)$ and defined in Equation (4.1), must satisfy

$$\mathcal{C}^{r^*,\mathrm{T},\alpha}_{<1,\infty>}(x,i) \leq \mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>}(x,i), \forall r \in R. \tag{4.6}$$

Proven theorems on dynamic programming and Markov decision processes allow us to extend the results from the finite horizon problem to an infinite horizon by extending the length of the planning horizon $N \to \infty$ ([7], [45], and [68]). Since the cost functions are stationary over time, we can drop the subscript t from Equation (3.13) and express the optimality equations for the infinite horizon problem as

$$\mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>}(x,i) = \inf_{y \leq x} \left\{ K(x-y) + G(y,i) + \mathcal{E}\mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>}(y,i) \right\}. \tag{4.7}$$

$G(y, i)$ is still the relevant cost per period, given by

$$G(y,i) = H(y) - (1-\alpha)C(y) + \sum_{q=0}^{Q}\sum_{j=1}^{M} C(q)[D_q]_{i,j}. \tag{4.8}$$

Condition 4.2.1, whose interpretation for the cost functions is similar to that of Condition 3.2.1, is required to establish our infinite-horizon results.

Condition 4.2.1.

(i) $K(w) = \hat{k} \cdot \delta_{\{|w|>0\}}$, *i.e., the fixed clearing cost is stationary over time and zero if nothing is cleared.*

(ii) $C(w) = \hat{c} \cdot |w|$, *i.e., the variable clearing cost is a linear function of the cleared quantities.*

(iii) $H(x+y) \geq H(x) + H(y)$, *for all* $x, y \in \Phi$, *i.e.,* H *is superadditive in* Φ.

(iv) $H(x \oplus [0, \ldots, 0] \oplus z) \geq H(x \oplus z)$, *for all* $x, y, z \in \Phi$, *i.e., delay penalties are non-decreasing over time.*

(v) $H(x) - (1-\alpha)C(x) \geq 0, \forall x \in \Phi$, *i.e., The variable clearing cost reduction from delaying clearance cannot exceed the corresponding delay penalty.*

(vi) For any real number $M > 0$, *there are only finitely many sequences* $x \in \Phi$ *such that* $H(x) - (1-\alpha)C(x) \leq M$.

(vii) $H_t(x) \geq H_t(\emptyset) = 0$, *for all* $x \in \Phi$ *and* $t \in T$.

To prove the existence of an optimal clearing policy, we need to show that there exists a solution to Equation (4.7) for the value function of the infinite horizon problem. Our method here is that of successive approximation of the infinite horizon problem by longer and longer finite-horizon problems.

First, let us examine the "first-n-period truncation" of the infinite horizon problem. Denote the expected total discounted cost for the truncated problem as

$$\mathcal{C}^{r,\mathrm{T},\alpha}_{<1,n>}(x,i) = E[\sum_{t=1}^{n} \alpha^{t-1} u^r(x_t, i_t)], \tag{4.9}$$

for all $(x,i) \in \Omega$. Next, define the value function for the truncated problem as

$$V_n(x,i) = \inf_{r \in R} \mathcal{C}^{r,\mathrm{T},\alpha}_{<1,n>}(x,i). \tag{4.10}$$

Naturally, the value function for this n-stage finite horizon problem exists, and can be computed recursively as

$$\begin{aligned} V_n(x,i) &= \min_{y \le x} \{K(x-y) + G(y,i) + \alpha \mathcal{E} V_{n-1}(y,i)\}, \\ V_0(x,i) &= 0, \end{aligned} \tag{4.11}$$

By definition, we have

$$\mathcal{C}^{r,\mathrm{T},\alpha}_{<1,\infty>}(x,i) = \lim_{n \to \infty} \mathcal{C}^{r,\mathrm{T},\alpha}_{<1,n>}(x,i), \tag{4.12}$$

for all $(x,i) \in \Omega$. In Proposition 4.1.1, we have already proven that the limit on the RHS exists. That is to say, the truncated problem converges to the infinite horizon problem.

According to Equation (4.6), if an optimal clearing rule r^* exists, then we have

$$\mathcal{C}^{r^*,\mathrm{T},\alpha}_{<1,\infty>}(x,i) \le \lim_{n \to \infty} \mathcal{C}^{r,\mathrm{T},\alpha}_{<1,n>}(x,i), \forall r \in R. \tag{4.13}$$

Now define the value function of the infinite horizon problem as

$$V(x,i) = \lim_{n \to \infty} \inf_{r \in R} \mathcal{C}^{r,\mathrm{T},\alpha}_{<1,n>}(x,i). \tag{4.14}$$

Since $\inf_{r\in R} \mathcal{C}^{r,\mathrm{T},\alpha}_{<1,n>}(x,i) = V_n(x,i)$, if we can prove that $V_n(x,i)$ converges as well, then we know that the value function for the infinite horizon problem exists, for which we have

$$V(x,i) = \lim_{n\to\infty} V_n(x,i) = \lim_{n\to\infty} \mathcal{C}^{r^*,\mathrm{T},\alpha}_{<1,n>}(x,i) = \mathcal{C}^{r^*,\mathrm{T},\alpha}_{<1,\infty>}(x,i). \tag{4.15}$$

Lemma 4.2.1. *Under Condition 4.2.1, we have*

$$V_n(x,i) \leq V_{n+1}(x,i) \tag{4.16}$$

for all $(x,i) \in \Omega$ *and* $n = 0, 1, \ldots$.

Proof. The proof is by induction on n. Since $V_0(x,i) = 0$ by definition, we then have

$$V_1(x,i) = \min_{y\leq x} \{K(x-y) + G(y,i)\} \geq 0,$$

because Condition 4.2.1.(v) ensures $G(y,i) \geq 0$. Assuming that inequality (4.16) holds for all $n \leq l$, we now must show that it also holds for $l+1$. Note that

$$\begin{aligned}
&V_{l+1}(x,i) - V_l(x,i) \\
&= \min_{y\leq x} \{K(x-y) + G(y,i) + \alpha\mathcal{E}V_l(y,i)\} - \min_{y\leq x} \{K(x-y) + G(y,i) + \alpha\mathcal{E}V_{l-1}(y,i)\} \\
&\geq \min_{y\leq x} \{K(x-y) + G(y,i) + \alpha\mathcal{E}V_{l-1}(y,i)\} - \min_{y\leq x} \{K(x-y) + G(y,i) + \alpha\mathcal{E}V_{l-1}(y,i)\} \\
&= 0.
\end{aligned}$$

This completes our proof. □

Next, we shall establish an upper bound on $V_n(x,i)$. One such possible bound is for the case that the maximum input quantity is received in each period, and the system is also cleared that often. A clearing cost of $(\hat{k} + \hat{c}Q)$ is thus incurred every period; no delay

penalty is ever charged since inputs are cleared at the earliest possible time. For this case,

$$\sup_{r\in R}\mathcal{C}^{r,\mathrm{T},\alpha}_{<1,n>}(x,i)\leq K(x)+C(x)+\sum_{t=2}^{n}\alpha^{t-1}(\hat{k}+\hat{c}Q),$$

and if $|x|<\infty$, we then have

$$\lim_{n\to\infty}\sup_{r\in R}\mathcal{C}^{r,\mathrm{T},\alpha}_{<1,n>}(x,i)\leq K(x)+C(x)+\frac{\alpha(\hat{k}+\hat{c}Q)}{1-\alpha}<\infty,$$

because $0\leq\alpha<1$.

Theorem 4.2.2. *Under Condition 4.2.1, the* $\lim_{n\to\infty}V_n(x,i)$ *exists, and*

$$V(x,i)=\lim_{n\to\infty}V_n(x,i), \tag{4.17}$$

which is a solution to the optimality Equations (4.7), for all $(x,i)\in\Omega$.

Proof. We first show that the limit exists. Lemma 4.2.1 and the existence of the upper bound for V_n suggest that the sequence of functions V_n is point-wise non-decreasing in n and bounded from above, i.e.,

$$0=V_0\leq V_1\leq\ldots\leq V_n\leq K(x)+C(x)+\frac{\alpha(\hat{k}+\hat{c}Q)}{1-\alpha}.$$

Therefore, by the Monotone Convergence Theorem, V_n converges point-wisely to V. We now show that this limit is a solution to Equation (4.7). First we have

$$V_n(x,i)\leq V_{n+1}(x,i)=\min_{y\leq x}\left\{K(x-y)+G(y,i)+\alpha\mathcal{E}V_n(y,i)\right\}.$$

Thus, as $n\to\infty$, both sides of the above inequality converge, so that

$$V(x,i)\leq\min_{y\leq x}\left\{K(x-y)+G(y,i)+\alpha\mathcal{E}V(y,i)\right\}.$$

On the other hand,

$$V_n(x,i) \geq V_{n-1}(x,i) = \min_{y \leq x} \{K(x-y) + G(y,i) + \alpha \mathcal{E} V_{n-2}(y,i)\}.$$

Passing to the limit in the preceding inequality yields

$$V(x,i) \geq \min_{y \leq x} \{K(x-y) + G(y,i) + \alpha \mathcal{E} V(y,i)\}.$$

This completes our proof. □

4.3 Characterizing the Optimal Policies

The lemmas and propositions obtained in Sections 3.3 and 3.4 for the finite horizon problem continue to hold for the infinite horizon problem. We summarize them with the following lemmas and propositions.

Proposition 4.3.1. *Under Condition 4.2.1, if* $x' \leq x$

$$V(x',i) \leq V(x,i) \leq V(x',i) + \hat{k}, \tag{4.18}$$

and

$$V(x,i) = \min \{K(x) + G(\emptyset,i) + \alpha \mathcal{E} V(\emptyset,i), G(x,i) + \alpha \mathcal{E} V(x,i)\}, \tag{4.19}$$

for all $i \in I$ *and* $0 \leq q \leq Q$.

Proof. First, with $z = x - x'$, it is easy to observe that

$$\begin{aligned}
&G(x,i) - G(x',i) \\
&= -[C(x) - C(x')] + [H(x) - H(x')] + \alpha \sum_{q=0}^{Q} \sum_{j=1}^{M} [C(x \oplus q) - C(x' \oplus q)] [D_q]_{i,j} \\
&\geq H(z) - (1-\alpha) C(z) \geq 0.
\end{aligned}$$

Next we see that V_n, the value function for the first-n-period truncated problem, satisfies Lemmas 3.3.2 and 3.3.4. Therefore, we must have

$$V_n(x', i) \leq V_n(x, i) \leq V_n(x', i) + \hat{k}.$$

Passing to the limit in the inequalities completes the proof. □

The following theorem is a direct result of Equation (4.19) in Proposition 4.3.1.

Theorem 4.3.2. *Under Condition 4.2.1, the optimal clearing policy exists and must be a on-off control clearing policy, i.e., if a clearing is triggered then all accumulated inputs must be cleared, such that*

$$V(x, i) = \min \{K(x) + W(\emptyset, i), W(x, i)\}, \tag{4.20}$$

where $W(x, i) = G(x, i) + \alpha \mathcal{E} V(x, i)$.

Applying the limit argument allows us to prove the other results listed below; hence we omit the proofs here.

Proposition 4.3.3. *Under Condition 4.2.1*

$$V(x \oplus z, i) \leq V(x \oplus y \oplus z, i) \leq V(x \oplus z, i) + \hat{k}, \tag{4.21}$$

for all $x, y, z \in \Phi$, $i \in I$, *and* $0 \leq q \leq Q$.

Proposition 4.3.4. *Suppose that Condition 3.2.1 holds. For any system state* $(x, i) \in \Omega$, *if* $H(x) - (1 - \alpha)C(x) < (1 - \alpha)\hat{k}$, *then the optimal action is not to clear* x; *on the other hand, if* $H(x) - (1 - \alpha)C(x) > \hat{k}$, *then the optimal action is to clear* x.

Lemma 4.3.5. *Under Condition 4.2.1, for every* $i \in I$, *there are only finitely many sequences which do not require clearing under the optimal policy.*

Theorem 4.3.6. *Under Condition 4.2.1, an MD optimal policy is always logical and feasible, i.e., it satisfies Condition 2.2.1 and 2.2.2, and the set $\Psi_{r_t^*}$ has a tree structure and finite size.*

Proposition 4.3.7. *Under Condition 4.2.1, if the delay penalty function is either purely quantity-dependent or purely age-dependent, and if $C(x) = 0$, then $H(x) \leq H(y)$ implies*

$$V(x,i) \leq V(y,i) \leq V(x,i) + \hat{k}, \tag{4.22}$$

for all $x, y \in \Phi$, $i \in I$, $t \in T$, and $0 \leq q \leq Q$.

Definition 4.3.1. *A state-dependent threshold clearing policy in an infinite horizon is denoted as $\pi^\tau = (r^\tau, \ldots)$. The clearing rule r^τ is determined by a nonnegative vector τ, for which we have*

$$r^\tau(x,i) = \begin{cases} 0, & \text{if } H(x) \leq \tau(i) \\ 1, & \text{otherwise} \end{cases}, \tag{4.23}$$

for all $(x,i) \in \Omega$.

Theorem 4.3.8. *Under Condition 4.2.1, if the delay penalty function is either purely quantity-dependent or purely age-dependent and $C(x) = 0$, then the optimal clearing policy can be a state-dependent threshold clearing policy denoted by a nonnegative vector τ^*. More specifically, there exists a non-empty sequence $x^*(i)$ such that*

$$x^*(i) = \arg\max_{x \in \Phi} \{G(x,i) : G(x,i) + \mathcal{E}V(x,i) \leq K(x) + G(\emptyset,i) + \mathcal{E}V(\emptyset,i)\}, \tag{4.24}$$

and

$$\tau^*(i) = H(x^*(i)), \tag{4.25}$$

for all $i \in I$.

Corollary 4.3.9. *If Condition 4.2.1 is satisfied and the delay penalty function is purely quantity-dependent, then the optimal clearing policy can be a state-dependent quantity*

threshold clearing policy denoted by a nonnegative vector Q^*. More specifically, there exists a non-empty sequence $x^*(i)$ such that

$$x^*(i) = \arg\max_{x\in\Phi}\{|x| : G(x,i) + \mathcal{E}V(x,i) \leq K(x) + G(\emptyset,i) + \mathcal{E}V(\emptyset,i)\}, \tag{4.26}$$

and

$$Q^*(i) = |x^*(i)|, \tag{4.27}$$

for all $i \in I$.

4.4 Computing the Optimal Policy Parameters

The infinite horizon stochastic clearing problem can be solved as an infinite horizon discounted *Markov decision process* (MDP). We can use value iteration or policy iteration to calculate its value function and optimal policy parameters.

Before constructing the computational algorithms, we shall divide the system state space Ω into two subsets, Ω_{U} and Ω_{C}. These subsets are defined as

$$\Omega_{\mathrm{U}} = \{(x \oplus q, i) \in \Omega : H(x) - C(x) + \alpha C(x) \leq \hat{k}\}, \tag{4.28}$$

and

$$\Omega_{\mathrm{C}} = \Omega \setminus \Omega_{\mathrm{U}}. \tag{4.29}$$

Ω_{U} and Ω_{C} are mutually exclusive and collectively exhaustive. According to Proposition 4.3.4, it is always optimal to clear any state $(x,i) \in \Omega_{\mathrm{C}}$, and if that is the initial system state, we can simply add $K(x) + C(x)$ to the total discounted cost and start from $(\emptyset, i)$ instead. Therefore, our algorithms do not need to record anything for states in Ω_{C}. On the other hand, Ω_{U} contains all remaining states for which the optimal decision is unknown. Therefore, it is for those states in Ω_{U} that our algorithm must determine the

best course of action. Lemma 4.3.5 has already confirmed that Ω_{U} is finite.

Let B denote the set of bounded real-valued functions on Ω_{U}. We shall define the norm of $u \in U$ by

$$||u|| = \sup_{(x,i)\in\Omega_{\mathrm{U}}} |u(x,i)|. \tag{4.30}$$

Since the set Ω_{U} is discrete and finitely large, B is a Banach space, i.e., a complete normed space.

Let us introduce an operator $\mathcal{T} : B \to B$ as

$$\mathcal{T}b(x,i) = \min\{G(x,i) + \alpha\mathcal{E}b(x,i), K(x) + G(\emptyset,i) + \alpha\mathcal{E}b(\emptyset,i)\}. \tag{4.31}$$

We can define another similar operator $\mathcal{T}_r : B \to B$ as

$$\mathcal{T}_r b(x,i) = \{K(y_r) + G(y_r,i) + \alpha\mathcal{E}b(y_r,i)\}, \tag{4.32}$$

where r is a given clearing rule and $y_r = x(1 - r(x,i))$.

4.4.1 Value Iteration Algorithm

The following value iteration algorithm uses the tree structure and Proposition 4.3.4 to help construct the optimal policy.

Algorithm IV: Discounted Cost Value Iteration Algorithm

IV.1 If $H(x) - (1-\alpha)C(x) > \hat{k}$, let $r(x,i) = 1$; otherwise, let $r(x,i) = 0$. Use BFT Procedure I to create Ω_{U}.

IV.2 Set $V^{[0]}(x,i) = 0$, for all $(x,i) \in \Omega_{\mathrm{U}}$. Choose an appropriate $\epsilon > 0$ and set counter $t = 0$.

IV.3 For each sequence $(x,i) \in \Omega_{\mathrm{U}}$, $j \in I$, and $q = 0, 1, \ldots, Q$, creates $y = x \oplus q$ and look up $V^{[t]}(y,j)$. If it cannot be found, set $V^{[t]}(y,j) = K(y) + V^{[t]}(\emptyset, j)$.

IV.4 Compute $V^{[t+1]}(x,i) = \mathcal{T}V^{[t]}(x,i)$ for all $(x,i) \in \Omega_{\mathrm{U}}$.

IV.5 If $||V^{[t+1]} - V^{[t]}|| < \frac{\epsilon(1-\alpha)}{2\alpha}$, go to Step 6; otherwise, increment t by 1 and go back to Step 3.

IV.6 If $G(x,i) + \alpha\mathcal{E}V^{[t+1]}(x,i) > K(x) + G(\emptyset,i) + \alpha\mathcal{E}V^{[t+1]}(\emptyset,i)$, set $r^*(x,i) = 1$; otherwise, set $r^*(x,i) = 0$. Lastly, compute V by running Algorithm III with respect to r^*.

The following proposition is essential to prove the convergence of Algorithm IV.

Proposition 4.4.1. *$\mathcal{T}$ and $\mathcal{T}_r$ are both contraction mappings, i.e., $||\mathcal{T}u - \mathcal{T}v|| \leq \alpha||u-v||$, for all $u, v \in B$ and some $0 \leq \alpha < 1$.*

Proof. Let vectors $u, v \in B$, and assume that $\mathcal{T}u(x,i) \leq \mathcal{T}v(x,i)$ for some fixed state (x,i). With

$$y^*_{(x,i)} \in \arg\min_{y \in \{\emptyset, x\}} \{K(x-y) + G(y,i) + \alpha\mathcal{E}u(y,i)\},$$

then

$$\begin{aligned} 0 &\leq \mathcal{T}v(x,i) - \mathcal{T}u(x,i) \\ &\leq K(x - y^*_{(x,i)}) + G(y^*_{(x,i)}, i) + \alpha\mathcal{E}v(y^*_{(x,i)}, i) - K(x - y^*_{(x,i)}) - G(y^*_{(x,i)}, i) - \alpha\mathcal{E}u(y^*_{(x,i)}, i) \\ &= \alpha\mathcal{E}(v(y^*_{(x,i)}, i) - u(y^*_{(x,i)}, i)) \\ &\leq \alpha\mathcal{E}||v-u|| = \alpha||v-u||. \end{aligned}$$

Repeating the same argument in the case that $\mathcal{T}u(x,i) \geq \mathcal{T}v(x,i)$, we get

$$|\mathcal{T}u - \mathcal{T}v| \leq \alpha||u-v||$$

for all $(x, i) \in \Omega_{\mathrm{U}}$. Taking the supremum over (x, i) completes the proof for $\mathcal{T}$. The proof for $\mathcal{T}_r$ follows the same argument. □

The convergence of Algorithm IV is demonstrated by our next result.

Theorem 4.4.2. *For each pair $(x, i) \in \Omega_{\mathrm{U}}$, the series of $V^{[t]}(x, i)$ has the following properties*

(i) $V^{[t]}$ converges in norm to V, which is the value function of the optimal policy;

(ii) The algorithm converges after finitely many iterations;

(iii) The stationary policy given by r is ϵ-optimal, i.e., $||V^r - V|| \leq \epsilon$;

(iv) $||V^{[t+1]} - V|| < \frac{\epsilon}{2}$;

where ϵ is the precision factor used in Algorithm IV as the termination condition.

Proof. Since Step 4 can be rewritten as Equation (4.31), and the operator $\mathcal{T}$ is a contraction mapping on the Banach space B, the Banach Fixed-Point Theorem implies parts (i) and (ii). Now suppose that the algorithm terminates at iteration t, outputting policy r and the corresponding expected total discounted cost V^r. Then

$$||V^r - V|| \leq ||V^r - V^{[t+1]}|| + ||V^{[t+1]} - V||.$$

By definition, we know that $V^r = \mathcal{T}_r V^r$ and $\mathcal{T}_r V^{[t+1]} = \mathcal{T} V^{[t+1]}$. The first term on the RHS of the above inequality can thus be written as

$$\begin{aligned} ||V^r - V^{[t+1]}|| &= ||\mathcal{T}_r V^r - V^{[t+1]}|| \\ &\leq ||\mathcal{T}_r V^r - \mathcal{T} V^{[t+1]}|| + ||\mathcal{T} V^{[t+1]} - V^{[t+1]}|| \\ &= ||\mathcal{T}_r V^r - \mathcal{T}_r V^{[t+1]}|| + ||\mathcal{T} V^{[t+1]} - \mathcal{T} V^{[t]}|| \\ &\leq \alpha ||V^r - V^{[t+1]}|| + \alpha ||V^{[t+1]} - V^{[t]}||, \end{aligned}$$

since both operators $\mathcal{T}$ and $\mathcal{T}_r$ are contraction mappings on B. Rearranging the terms yields

$$||V^r - V^{[t+1]}|| \leq \frac{\alpha}{1-\alpha}||V^{[t+1]} - V^{[t]}||.$$

Using a similar argument, we can obtain

$$||V^{[t+1]} - V|| \leq \frac{\alpha}{1-\alpha}||V^{[t+1]} - V^{[t]}||.$$

The termination condition, $||V^{[t+1]} - V^{[t]}|| < \frac{\epsilon(1-\alpha)}{2\alpha}$, then ensures

$$||V^{[t+1]} - V|| < \frac{\epsilon}{2},$$

and

$$||V^r - V|| \leq \epsilon.$$

This completes the proof. □

Rate of Convergence: According Theorem 6.3.3 in [68], the convergence for the value iteration algorithm is linear at rate α.

Time Complexity: Similar to the time complexity of Algorithm I, the size of Ω_{U} can be estimated by $MQ^{\tilde{L}}$, where $\tilde{L}$ is given by Equation (3.40). In Step IV.6, Algorithm III is run with respect to Ω_{U}, so its complexity is $\mathcal{O}(M^3Q^{\tilde{L}})$. If we do not need to report V in Step IV.6, then this step has significantly lower complexity. The recursive steps IV.3 through IV.5 require traversing the entire set Ω_{U}, accessing stored values $V^{[t]}(x \oplus q, j)$ for each $(x, i) \in \Omega_{\mathrm{U}}$, $j \in I$, and $q = 0, 1, \ldots, Q$, then computing $V^{[t+1]}(x, i)$. In each iteration, these steps have complexity of $\mathcal{O}(M^2Q^{\tilde{L}})$ because the tree structure of Ω_{U} allows us to store links between $V^{[t]}(x, i)$ and $V^{[t]}(x \oplus q, j)$ directly.

Memory Complexity: Using Algorithm III in Step 6 has memory complexity of $\mathcal{O}(M^2Q^{\tilde{L}})$. In all other steps, the sequence x, $V^{[t]}(x, i)$ and $V^{[t+1]}(x, i)$ are stored for all $(x, i) \in \Omega_{\mathrm{U}}$

in each iteration t. Therefore, the combined memory complexity for the other steps is $\mathcal{O}(M\tilde{L}Q^{\tilde{L}})$.

Please refer to Appendix C for a numerical example to verify the complexity estimates.

4.4.2 Policy Iteration Algorithm

An alternative computation approach is through a modified policy iteration algorithm.

Algorithm V: Discounted Cost Policy Iteration Algorithm

V.1 If $H(x) - (1-\alpha)C(x) > \hat{k}$, let $r^{[0]}(x,i) = 1$; otherwise, let $r^{[0]}(x,i) = 0$. Use BFT Procedure I to create Ω_{U}. Set counter $t = 0$.

V.2 (Policy Evaluation) Run Algorithm II with respect to $r^{[t]}$ to compute $V^{[t]}(x,i)$, for all $(x,i) \in \Omega_{\mathrm{U}}$.

V.3 (Policy Improvement) For every $(x,i) \in \Omega_{\mathrm{U}}$, set $r^{[t+1]}(x,i) = 1$ if $G(x,i) + \alpha\mathcal{E}V^{[t]}(x,i) > K(x) + G(\emptyset,i) + \alpha\mathcal{E}V^{[t]}(\emptyset,i)$; otherwise, set $r^{[t+1]}(x,i) = 0$.

V.4 If $r^{[t+1]} = r^{[t]}$, stop and set $r^* = r^{[t+1]}$; otherwise increment t by 1 and return to Step 2.

The key to proving convergence of Algorithm V is

Proposition 4.4.3. *The successive values $V^{[t]}$ and $V^{[t+1]}$ generated by Algorithm V are non-increasing, i.e., $V^{[t]} \geq V^{[t+1]}$.*

Proof. Recall from Section 4.1 that Algorithm II computes $V^{[t]}$ by solving the following equation

$$V^{[t]} = u^{r^{[t]}} + \alpha P^{(x,r^{[t]})} V^{[t]},$$

where the matrix $P^{(x,r^{[t]})}$ is generated by Equations (2.25) to (2.30). Therefore, Step 3 can be expressed as

$$r^{[t+1]} \in \arg\min_{r \in R}\{u^r + \alpha P^{(x,r)} V^{[t]}\}.$$

Then we have

$$u^{r^{[t+1]}} + \alpha P^{(x,r^{[t+1]})} V^{[t]} \leq u^{r^{[t]}} + \alpha P^{(x,r^{[t]})} V^{[t]} = V^{[t]}.$$

Rearranging terms we get

$$u^{r^{[t+1]}} \leq (I - \alpha P^{(x,r^{[t+1]})}) V^{[t]}.$$

Pre-multiplying both sides by $(I - \alpha P^{(x,r^{[t+1]})})^{-1}$ yields

$$V^{[t+1]} = (I - \alpha P^{(x,r^{[t+1]})})^{-1} u^{r^{[t+1]}} \leq V^{[t]}.$$

This completes the proof. □

The following theorem demonstrates the convergence of Algorithm V.

Theorem 4.4.4. *Algorithm V terminates in a finite number of iterations with a solution of the optimality equation $V^{[t]}$ and an optimal policy r^*.*

Proof. Because of the finite nature of both the system state space Ω_{U} and the action set A, there can only be finitely many MD policies. We have already shown that an optimal policy can be an MD policy, and Proposition 4.4.3 suggests that $V^{[t]}$ is non-increasing. Therefore, the algorithm must terminate after a finite number of iterations. □

Rate of Convergence: According Theorem 6.4.6 in [68], the convergence for the policy iteration algorithm is at least linear. Under special conditions specified in Theorem 6.4.8 in [68], the convergence rate can be quadratic.

Time Complexity: The tree structure of Ω_{U} allows its size to be estimated by $MQ^{\tilde{L}}$, where $\tilde{L}$ is given by Equation (3.40). In the policy evaluation step, Algorithm II is run with

respect to Ω_{U}, so its complexity is $\mathcal{O}(M^3 Q^{\tilde{L}})$. The complexity of the policy improvement step is $\mathcal{O}(M^2 Q^{\tilde{L}})$ because the tree structure of Ω_{U} allows us to store links between $V^{[t]}(x,i)$ and $V^{[t]}(x \oplus q, j)$ directly.

Memory Complexity: The policy evaluation step has memory complexity of $\mathcal{O}(M^2 Q^{\bar{L}})$ because of Algorithm II. Whereas the policy improvement step has memory requirement complexity of $\mathcal{O}(M\tilde{L}Q^{\tilde{L}})$ because the sequence x, $V^{[t]}(x,i)$ and $V^{[t+1]}(x,i)$ are stored for all $(x,i) \in \Omega_{\mathrm{U}}$ in each iteration t.

Please refer to Appendix C for a numerical example to verify the complexity estimates. Note that the optimal clearing policy can also be found via a linear program of the corresponding Markov Decision Process. We modify the existing LP formulation in [68] to solve our version of stochastic clearing problem in Appendix A.

We now use Algorithms IV and V to solve the following stochastic clearing problem separately, i.e., by value iteration and then by policy iteration. Results of the two methods are found to be consistent (see Table 4.2).

Example 4.4.1. *Consider a stochastic clearing system with the following parameters.*

(i) The discount factor $\alpha = 0.95$.

(ii) The input process is given as the BMAP in Example 2.1.1.(iii).

(iii) The delay penalty cost is given as in Example 2.3.1.(d).

(iv) The fixed clearing cost is $K(w_t) = 5 \cdot \delta_{\{|w_t|>0\}}$.

(v) The variable clearing cost is $C(w_t) = 0.5|w_t|$.

The optimal clearing policy and the corresponding value functions are summarized in Table 4.2. The cost functions given in Example 4.4.1 clearly satisfy Condition 4.2.1. The numerical results of this example show that the optimal stationary clearing rules depend

Result Summary	$r^*(x,1)$	$r(x,2)$	$V(x,1)$	$V(x,2)$
$\emptyset$	0	0	4.0742	44.0103
$x=[1]$	0	0	9.2366	45.4745
$x=[2]$	1	1	10.0743	50.0103
$x=[3]$	1	1	10.5743	50.5103
$x=[1,0]$	1	0	9.5743	46.2245
$x=[1,1]$	1	1	10.0743	50.0103
$x=[1,2]$	1	1	10.5743	50.5103
$x=[1,3]$	1	1	11.0743	51.0103
$x=[2,0]$	1	1	10.0743	50.0103
$x=[2,1]$	1	1	10.5743	50.5103
$x=[1,0,0]$	1	1	9.5743	49.5103
$x=[1,0,1]$	1	1	10.0743	50.0103
$x=[1,0,2]$	1	1	10.5743	50.5103
$x=[1,0,3]$	1	1	11.0743	51.0103
$x=[1,1,0]$	1	1	10.0743	50.0103
$x=[1,1,1]$	1	1	10.5743	50.5103
$x=[1,1,2]$	1	1	11.0743	51.0103
$x=[1,0,0,0]$	1	1	9.5743	49.5103
$x=[1,0,0,1]$	1	1	10.0743	50.01034

Table 4.2: Summary of Optimal Clearing Rule for Example 4.4.1

on the underlying state i, and demonstrate the characteristics proven in Section 4.3. In Example 4.4.2, we show a stochastic clearing problem with an optimal state-dependent threshold clearing policy. In particular, note that for $x=[0,1,0]$, it is optimal to clear the system if the underlying state is 1 but it is optimal to continue to accumulate if the underlying state is 2. This is intuitive because there is a much higher input arrival rate in state 1 than state 2. Also note that for all the system content states that require clearing, the expected total discounted costs differ by the variable clearing costs only.

Example 4.4.2. *Consider a stochastic clearing system with the following parameters.*

(i) The planning horizon has three periods.

(ii) The input process is given as the BMAP in Example 2.1.1.(ii).

(iii) The delay penalty cost is given as in Example 2.3.1.(c).

(iv) The fixed clearing cost is $K(w_t) = 5 \cdot \delta_{\{|w_t|>0\}}$.

(v) The variable clearing cost is $C(w_t) = 0$.

x	$r^*(x,1)$	$r^*(x,2)$	$V(x,1)$	$V(x,2)$	$H(x,2)$
$\emptyset$	0	0	0.5468	0.1823	0
[1]	0	0	1.3163	0.9518	0.5
[2]	0	0	1.6808	1.3163	0.5
[1,0]	1	1	1.7618	1.3973	2
[1,1]	1	1	2.1668	1.8023	2.5
[1,2]	1	1	2.5718	2.2073	2.5
[2,0]	1	1	2.1668	1.8023	2
[2,1]	1	1	2.5718	2.2073	2.5
[2,2]	1	1	2.9768	2.6123	2.5

Table 4.3: Summary of Optimal Clearing Rule in Period 1 for Example 4.4.2

The optimal clearing policy and the corresponding value functions are summarized in Table 4.3. The cost functions given in Example 4.4.2 clearly satisfy both Conditions 4.2.1 and $C(w_t) = 0$, and H is purely-age-dependent. The numerical results of this example show that the optimal clearing rules can be converted into state-dependent threshold rules defined in Section 4.3. The optimal threshold policy parameters are found as $\tau^*(1) = \tau^*(2) = 0.5$.

4.5 Summary of Results

In this chapter, we first introduce an algorithm to compute the expected discounted cost over an infinite horizon for any given clearing policy. We then use a limit argument to extend the results for the expected total cost problem over a finite horizon to the expected discounted cost problem over an infinite horizon. More specifically, the ordering of the expected discounted cost is still induced by a partial order on the initial system state space; and the optimal clearing policy is an on-off control policy for which partial clearing is never allowed.

The above-mentioned characteristics of the optimal policy enable us to modify the value/policy iteration algorithms to search for the optimal clearing policy parameters. The modified algorithms are subsequently shown to have exponential complexities, but are still capable of solving real-life problems of moderate sizes.

Although the expected total cost or total discounted cost is the preferred measure of several applications of stochastic clearing systems, such as inventory control, the expected average cost per period may be more appropriate for other applications such as shipment consolidation. In the next two chapters, we will shift our focus to the expected average cost per period objective function.

Chapter 5

Expected Average Cost Model over Infinite Horizon (MDP)

In this chapter, we use the concept of Markov Reward Processes (MRP) to study the stochastic clearing problem with expected average total cost per period as the optimality objective. MRP is a variation of Markov Decision Processes (MDP), providing us with several ways to find the optimal policy parameters even if the structure of the optimal policy is unknown. The modelling assumptions and notations used in this chapter are summarized below.

(i) The planning horizon is infinite, i.e., $T = \{1, 2, \ldots\}$.

(ii) The model is in discrete time and discrete quantities, hence the input process is modelled as a *BMAP*.

(iii) The clearing policy is given by a stationary policy $\pi = (r, r, \ldots)$, thus π and r can be used interchangeably.

(iv) The pre-clearing system state process $\{(x_t, i_t), t = 1, 2, \ldots\}$ is the Markov chain of interest in this chapter.

(v) The cost functions are given by H, K, and C, which are stationary over time. (We can add the subscript t, if necessary, and extend the procedures below).

Note that the modelling assumptions and notations used in this chapter are almost the same as in Chapter 4. The only exception is that we disregard the discounting factor since we are interested in the long-run average total cost.

5.1 Policy Evaluation

We have already used the concepts of Markov decision processes to study the optimal clearing policies for total-cost and discounted-total-cost objectives in Chapters 3 and 4, respectively. MDP can be applied to problems with expected average total cost objective as well. In this case, we focus on the pre-clearing system state process $\{(x_t, i_t), t = 1, 2, \ldots\}$ and a stationary clearing policy $\pi = (r, r, \ldots)$.

5.1.1 Markov Reward Process

In the early chapters, we established that the stochastic clearing problem can be modelled as an MDP with a Markov chain $\{(x_t, i_t), t = 1, 2, \ldots\}$, stationary clearing rule r, transition probability matrix $P^{(x,r)}$ defined by Equations (2.25) to (2.27), and cost function $u^r(x, i)$ defined by Equation (3.1). From [68], we know that we can refer to the stochastic process $\{((x_t, i_t), u^r(x_t, i_t)), t = 1, 2, \ldots\}$ as a *Markov reward process* (MRP), where the reward is in fact the cost associated with the clearing process. To simplify our notation, we shall use P to represent $P^{(x,r)}$ in this chapter whenever it is possible.

We now apply the concepts of MRP to our model of stochastic clearing processes. The expected average reward function for MRP is denoted by a vector g^r for each clearing rule

$r \in R$, and defined as

$$g^r(x,i) = \lim_{N\to\infty} \frac{1}{N} E\left[\sum_{t=1}^{N} u^r(x_t, i_t)\right] = \lim_{N\to\infty} \mathcal{C}^{r,\mathrm{A}}_{<1,N>}(x,i), \tag{5.1}$$

which is actually equal to the expected average total cost per period.

Our objective in this chapter is to seek a method for computing

$$g^*(x,i) = \inf_{r\in R} g^r(x,i) \tag{5.2}$$

and finding a policy $r^* \in R$ for which

$$g^{r^*}(x,i) = g^*(x,i), \tag{5.3}$$

for all $(x,i) \in \Omega$. It turns out that this approach has some shortcomings because the limits in Equation (5.1) may not always exist.

If P is stochastic and aperiodic, then it has already been proven in [68] that the limits in Equation (5.1) exist and

$$g^r = P^* u^r, \tag{5.4}$$

where P^* is the *limiting matrix*, i.e., each row of P^* is identical and contains the steady state probabilities of P. According to the theory of discrete time Markov chains, this limiting matrix can be computed by

$$P^* = \lim_{N\to\infty} P^N, \tag{5.5}$$

if P is aperiodic, or calculated as the Cesaro limit

$$P^* = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} P^n, \tag{5.6}$$

if P is periodic, where P^n is the *n-step transition probability matrix* of $\{(x_t, i_t), t = 1, 2, \ldots\}$, and computed as P to the power of n.

On the other hand, we can define another function g, known as the "gain" of the MRP, by

$$g = P^* u^r. \tag{5.7}$$

Note that g^r and g are not exactly the same because g^r may not exist at all. Hence, we focus on computing g instead.

Since, according to Theorem 2.2.3, P is stochastic, irreducible, and aperiodic, $g(x, i)$ is thus proven to be a constant function (see [68]). Therefore, we can rewrite Equation (5.7) in the matrix notation

$$\hat{g}\boldsymbol{e} = P^* u^r, \tag{5.8}$$

where $\hat{g}$ is the constant value of g.

The "bias" of MRP, denoted by function h, is given as

$$h(x, i) = E\left[\sum_{t=1}^{\infty} (u^r(x_t, i_t) - g(x_t, i_t)) \,\middle|\, (x_t, i_t) = (x, i)\right]. \tag{5.9}$$

According to [68], we can write Equation (5.9) in the matrix form as

$$h = H_P u^r, \tag{5.10}$$

where H_P is the *deviation matrix* defined by

$$H_P = (I - P + P^*)^{-1}(I - P^*). \tag{5.11}$$

Note that H_P is also known as the *Drazin inverse* of $I - P$.

Equations (5.10) and (5.11) gives us an alternative interpretation for the bias of MRP,

that is

$$h = \sum_{t=1}^{N} P^{t-1}u^r - N\hat{g}\boldsymbol{e} + \sum_{t=N+1}^{\infty} (P^{t-1} - P^*)u^r, \tag{5.12}$$

where the third term above converges to zero as $N \to \infty$. We can easily see that

$$\sum_{t=1}^{N} P^{t-1}u^r = E\left[\sum_{t=1}^{N} u^r(x_t, i_t)\right] = \mathcal{C}^{r,\mathrm{T}}_{<1,N>}, \tag{5.13}$$

which is the vector of expected total costs in the first N periods for any initial system state. We thus have

$$\mathcal{C}^{r,\mathrm{T}}_{<1,N>} = h + N\hat{g}\boldsymbol{e} + o(1), \tag{5.14}$$

where $o(1)$ is a vector with elements approaching zero pointwise as $N \to \infty$. Therefore, h can be thought as the "intercept" of the N-period expected total cost function and g can be regarded as the "slope". h captures the impact of the initial system state on the total cost. $\hat{g}$ represents the long-run growth rate of the costs once the system enters the steady state, which is independent of the initial system state.

For any pair of system states (x, i) and (y, j), Equation (5.14) suggests that

$$h(x, i) - h(y, j) = \lim_{N\to\infty} [\mathcal{C}^{r,\mathrm{T}}_{<1,N>}(x, i) - \mathcal{C}^{r,\mathrm{T}}_{<1,N>}(y, j)]. \tag{5.15}$$

Hence, h is also the asymptotic relative difference in expected total cost that results from starting the process in state (x, i) instead of in state (y, j). [68] also refers to the vector h as the *relative value* vector.

5.1.2 Policy Evaluation Equations

Although $\hat{g}$ and h can be computed through direct evaluation of P^* and H_P, that can be inefficient in some cases. This is because the size of the state space for $\{(x_t, i_t), t = 1, 2, \ldots\}$ may be quite large, therefore making it difficult to compute P^* and H_P without the presence

of any special structure. The following computational method may be more efficient to some extent.

Theorem 5.1.1. *Suppose that the pre-clearing system state $\Lambda_{(r)}$ is finite. The gain $\hat{g}$ and bias h of the MRP with irreducible transition probability matrix P and reward u^r then satisfy the equation*

$$g + (I - P)h = u^r. \tag{5.16}$$

Proof. Since the Markov chain of P is stochastic and irreducible, g is constant across all underlying states, and can be represented by $\hat{g}\boldsymbol{e}$. The rest of the proof follows directly from Theorem 8.2.6 and Corollary 8.2.7 in [68]. □

Note that Equation (5.16) is actually a system of $|\Lambda_{(r)}|$ equations and $|\Lambda_{(r)}|+1$ variables, which means we have one "free" variable. Since h is the asymptotic relative difference between two system states, we can actually set one of them to zero as the baseline for comparison. In this way, we now have a system of $|\Lambda_{(r)}|$ equations with $|\Lambda_{(r)}|$ variables, which can be solved by standard linear algebra methods such as Gaussian elimination.

We employ Algorithm VI to compute the expected average total cost per period over the infinite horizon.

Algorithm VI: Infinite Horizon Average Cost Evaluation Algorithm

VI.1 Construct $\Psi_{(r)}$ using BFT Procedure I.

VI.2 For each $x \in \Psi_{(r)}$ and $q = 0, 1, \ldots, Q$, obtain matrices $A^{(x,r)}_{x,x\oplus q}$ and $B^{(x,r)}_{x,q}$ by Equations (2.29) and (2.30), and form the transition probability matrix P; also compute $u^r(\boldsymbol{x})$ by Equations (3.1) and (4.3), and form the vector u^r.

VI.3 Set $h(\emptyset, 1) = 0$, and solve Equation (5.16) by Gaussian elimination.

Time Complexity: BFT Procedure I has time complexity $\mathcal{O}(MQ^{\bar{L}(r)})$. The computation time in Step VI.2 scales by $\mathcal{O}(MQ^{\bar{L}(r)})$ because we only need to compute $r(x \oplus q, j)$ for each x, q and j to form block matrices A and B. Finally, Step VI.3 requires solving a system of equations of size $MQ^{\bar{L}(r)}$. The Gaussian elimination method has a time complexity of $\mathcal{O}(M^3Q^{\bar{L}(r)})$. Overall, the time complexity of Algorithm VI is $\mathcal{O}(M^3Q^{\bar{L}(r)})$.

Memory Complexity: The greatest memory is required to store the matrix P, whose size is bounded by $MQ^{\bar{L}(r)} \times MQ^{\bar{L}(r)}$. Therefore, the memory requirement complexity is $\mathcal{O}(M^2Q^{\bar{L}(r)})$.

Example 5.1.1. *Consider 12 different stochastic clearing problems with the following parameters.*

(i) Three distinct input processes described in Example 2.1.1.

(ii) Four separate delay penalty cost functions H, as in Example 2.3.1.

(iii) The fixed clearing cost of the form $K(w_t) = 10 \cdot \delta_{\{|w_t|>0\}}$.

(iv) The variable clearing cost is expressed as $C(w_t) = 0.5|w_t|$.

(v) The clearing policy is given as

$$r(x_t, i_t) = \begin{cases} 0, & \text{if } |x_t| < 5 \text{ and } \mathcal{L}(x_t) < 4 \\ 1, & \text{otherwise} \end{cases}.$$

The expected average costs for the various problems are computed by Algorithm VI and summarized in Table 5.1.

$H \backslash BMAP$	Example 2.1.1.(i)	Example 2.1.1.(ii)	Example 2.1.1.(iii)
Example 2.3.1.(ii)	1.8364	1.3964	1.6622
Example 2.3.1.(iii)	1.8070	1.4051	1.6293
Example 2.3.1.(iv)	1.4761	1.1302	1.4120

Table 5.1: Summary of Expected Average Costs for Example 5.1.1

5.2 Optimality Equation

In this section, we set up the optimality equation for the expected average total cost model, then prove the existence of an optimal clearing policy. Condition 5.2.1 on the cost functions are required to establish our optimality results for the given model. Interpretation of these conditions is similar to that of Condition 3.2.1.

Condition 5.2.1.

(i) $K(w) = \hat{k} \cdot \delta_{\{|w|>0\}}$, *i.e., the fixed clearing cost is stationary over time and zero if nothing is cleared.*

(ii) $C(w) = \hat{c} \cdot |w|$, *i.e., the variable clearing cost is a linear function of the cleared quantities.*

(iii) $H(x+y) \geq H(x) + H(y)$, *for all* $x, y \in \Phi$, *i.e.,* H *is superadditive in* Φ.

(iv) $H(x \oplus [0, \ldots, 0] \oplus z) \geq H(x \oplus z)$, *for all* $x, y, z \in \Phi$, *i.e., delay penalties are non-decreasing over time.*

(v) For any real number $M > 0$, *there are only finitely many sequences* $x \in \Phi$ *such that* $H(x) \leq M$.

Recall that for a given clearing policy r, the expected average cost per period over the infinite horizon is given by

$$\mathcal{C}^{r,\mathrm{A}}_{<1,\infty>} = \lim_{N\to\infty} \frac{1}{N} \mathcal{C}^{r,\mathrm{T}}_{<1,N>} = g,$$

and the gain and bias satisfy

$$g + (I - P)h = u^r.$$

The following lemma shows the optimal clearing decision for some system states.

Lemma 5.2.1. *Under Condition 5.2.1, if $H(x_t) > \hat{k}$ in period t, it is optimal to clear the system immediately.*

Proof. Let us consider two identical clearing systems. At decision epoch t, we choose to continue to accumulate in System 1, and use it as the benchmark. For System 2, let us clear all outstanding inputs at t, then mimic the decisions of System 1 regardless of the state of system 2. After the next clearing takes place in System 1 at some future decision epoch $n > t$, both systems will become identical again. Now suppose $H(x_t) > \hat{k}$ for System 1; Conditions 5.2.1.(iii) and (iv) imply that at any future decision epoch $j > t$, $H(x_j) \geq H(x_t) > \hat{k}$. It is then easy to see that System 2 incurs lower cost than System 1 between period t and period n, but in any other period, both systems have the same cost. Therefore, System 2 has a lower average total cost per period. This means whenever $H(x_t) > \hat{k}$, we can choose to clear the system and start from an empty state to reduce costs. This is similar to the idea of Lemma 3.3.4, but in the context of expected average total cost per period. □

Similar to the discussion of Section 4.4, Lemma 5.2.1 allows us to divide the system state space Ω into two subsets, Ω_{U} and Ω_{C}. They are defined as

$$\Omega_{\mathrm{U}} = \{(x \oplus q, i) \in \Omega : H(x) \leq \hat{k}\}, \tag{5.17}$$

and

$$\Omega_{\mathrm{C}} = \Omega \setminus \Omega_{\mathrm{U}}. \tag{5.18}$$

These subsets are mutually exclusive and collectively exhaustive. Again, let B denote the set of bounded real-valued functions on Ω_{U}.

Proposition 5.2.2. *Under Condition 5.2.1, the state space of the MRP problem can be as small as Ω_{U}.*

Proof. According to Lemma 5.2.1, for any system state $(x, i) \in \Omega_{\mathrm{C}}$, the optimal clearing decision is to clear everything and start again from the empty state. If any such state (x, i) is given as the initial system state, Condition 5.2.1 ensures that no such state will ever be reached again in the future. The one-time clearing cost becomes negligible in the average cost calculation. Therefore, we only need to determine the optimal clearing decisions for the other system states in Ω_{U}. □

According to [68], the optimality equation in an average reward MDP may be expressed in matrix-vector notation as

$$0 = \min_{r \in R} \left\{ u^r - g + (P^{(x,r)} - I)h \right\} \equiv \mathcal{B}(g, h), \tag{5.19}$$

where $\mathcal{B}$ is a mapping from $\mathbb{R} \times B$ to B, and B is the the set of bounded real-valued functions on Ω_{U}.

However, if no partial clearing is allowed, recall from Equation (3.11) that we can write

$$u^r(x, i) = \begin{cases} G(x, i), & \text{if } r(x, i) = 0; \\ K(x) + G(\emptyset, i), & \text{otherwise.} \end{cases} \tag{5.20}$$

Then we can simplify Equation (5.19) as

$$0 = \min \left\{ \begin{array}{l} G(x, i) - g(x, i) - h(x, i) + \sum_{q=0}^{Q} P^{(x,r)}_{(x,i),(x \oplus q, j)} h(x \oplus q, j), \\ \\ K(x) + G(\emptyset, i) - g(x, i) - h(x, i) + \sum_{q=0}^{Q} P^{(x,r)}_{(x,i),(q,j)} h(q, j) \end{array} \right\}, \tag{5.21}$$

for all $(x, i) \in \Omega_{\mathrm{U}}$.

One of the most important results on the average reward of an MDP is presented here.

Theorem 5.2.3. *(Theorem 8.4.1 in [68]) Since Ω_{U} is finite, if there exists a scalar $\hat{g}$ and*

an $h \in B$ for which $\mathcal{B}(\hat{g}\boldsymbol{e}, h) = 0$, then

$$\hat{g}\boldsymbol{e} = g^*, \tag{5.22}$$

which is the minimum expected average total cost per period.

5.3 Computing the Optimal Policy Parameters

The existence and uniqueness of such a $\hat{g}$ and h have been proven in [68], and they are marked as $\hat{g}^*$ and h^*. Now, we can use the concept of *h-improving clearing rules* to construct the optimal clearing rule r^* from $\hat{g}^*$ and an h^*.

According to [68], a decision rule r_h is called h-improving if

$$r_h \in \arg\min_{r \in R}\{u^r + P^{(x,r)}h\}. \tag{5.23}$$

Then for finite Ω_{U} and a finite action set A, it has been proven that any stationary policy derived from an h^*-improving clearing rule is average optimal (Theorem 8.4.5 in [68]). This enables us to use value iteration, policy iteration, or linear programming to find the optimal clearing policy. We shall explain each algorithm in detail in the next section.

5.3.1 Value Iteration Algorithm

The sequence of values generated by value iteration can be expressed as

$$v^{[t+1]} = \mathcal{T}_{\mathrm{avg}} v^{[t]}, \tag{5.24}$$

where

$$\mathcal{T}_{\text{avg}}v(x,i) \equiv \min \left\{ \begin{array}{l} G(x,i) + \mathcal{E}v(x,i), \\ K(x) + G(\emptyset,i) + \mathcal{E}v(\emptyset,i) \end{array} \right\}, \tag{5.25}$$

for all $(x,i) \in \Omega_{\text{U}}$.

We shall use the tree structure of Φ and Lemma 5.2.1 to create Ω_{U}, and then modify the value iteration algorthm for MRP as follows.

Algorithm VII: Average Cost Value Iteration Algorithm

VII.1 If $H(x) > \hat{k}$, let $r(x,i) = 1$; otherwise, let $r(x,i) = 0$. Use BFT Procedure I to create Ω_{U}.

VII.2 Set $v^{[0]}(x,i) = 0$, for all $(x,i) \in \Omega_{\text{U}}$. Choose an appropriate $\epsilon > 0$ and set counter $t = 0$.

VII.3 For each sequence $(x,i) \in \Omega_{\text{U}}$, $j \in I$, and $q = 0, 1, \ldots, Q$, create $y = x \oplus q$ and look up $v^{[t]}(y,j)$. If it cannot be found, set $v^{[t]}(y,j) = K(y) + v^{[t]}(\emptyset,j)$.

VII.4 Compute $v^{[t+1]}(x,i) = \mathcal{T}v^{[t]}(x,i)$ for all $(x,i) \in \Omega_{\text{U}}$.

VII.5 If $sp(v^{[t+1]} - v^{[t]}) < \epsilon$, go to Step 6; otherwise, increment t by 1 and go back to Step 3.

VII.6 If $G(x,i) + \mathcal{E}v^{[t+1]}(x,i) > K(x) + G(\emptyset,i) + \mathcal{E}v^{[t+1]}(\emptyset,i)$, set $r^*(x,i) = 1$; otherwise, set $r^*(x,i) = 0$. Lastly, compute v^* by running Algorithm VI with respect to r^*.

Note that we use the span seminorm $sp(v)$ instead of the norm $||v||$ to check the termination condition in Step 5. More specifically, we have

$$sp(v) \equiv \max_{(x,i)\in\Omega_{\text{U}}} v(x,i) - \min_{(x,i)\in\Omega_{\text{U}}} v(x,i). \tag{5.26}$$

This is necessary for ensuring the convergence of the algorithm and that r^* is ϵ-optimal. For more details, please refer to Theorems 8.5.2 to 8.5.6 in [68].

Rate of Convergence: According Theorem 8.5.4 in [68], the value iteration algorithm converges after finitely many iterations.

Time Complexity: Similar to the time complexity of Algorithm I, the size of Ω_{U} can be estimated by $MQ^{\tilde{L}}$, where $\tilde{L}$ is given by Equation (3.40). In Step VII.6, Algorithm V is run with respect to Ω_{U}, so its complexity is $\mathcal{O}(M^3Q^{\tilde{L}})$. If we do not need to report v^* in Step VII.6, then this step will have significantly lower complexity. The recursive steps VII.3 through VII.5 require traversing the entire set Ω_{U}; accessing stored values $v^{[t]}(x \oplus q, j)$ for each $(x, i) \in \Omega_{\mathrm{U}}$, $j \in I$, and $q = 0, 1, \ldots, Q$; then computing $v^{[t+1]}(x, i)$. In each iteration, these steps have complexity of $\mathcal{O}(M^2Q^{\tilde{L}})$ because the tree structure of Ω_{U} allows us to store links between $v^{[t]}(x, i)$ and $v^{[t]}(x \oplus q, j)$ directly.

Memory Complexity: Using Algorithm VI in Step VII.6 has memory complexity of $\mathcal{O}(M^2Q^{\tilde{L}})$. In all other steps, x, $v^{[t]}(x, i)$ and $v^{[t+1]}(x, i)$ are stored for all $(x, i) \in \Omega_{\mathrm{U}}$ in each iteration t. Therefore, the combined memory complexity for the other steps is $\mathcal{O}(M\tilde{L}Q^{\tilde{L}})$.

5.3.2 Policy Iteration Algorithm

An alternative computational approach is through a modified policy iteration algorithm.

Algorithm VIII: Discounted Cost Policy Iteration Algorithm

VIII.1 If $H(x) > \hat{k}$, let $r^{[0]}(x, i) = 1$; otherwise, let $r^{[0]}(x, i) = 0$. Use BFT Procedure I to create Ω_{U}. Set counter $t = 0$.

VIII.2 (Policy Evaluation) Run Algorithm V with respect to $r^{[t]}$ to compute $g^{[t]}(x, i)$ and $h^{[t]}(x, i)$, for all $(x, i) \in \Omega_{\mathrm{U}}$.

VIII.3 (Policy Improvement) For every $(x,i) \in \Omega_{\mathrm{U}}$, set $r^{[t+1]}(x,i) = 1$ if $G(x,i) + \mathcal{E}h^{[t]}(x,i) > K(x) + G(\emptyset,i) + \mathcal{E}h^{[t]}(\emptyset,i)$; otherwise, set $r^{[t+1]}(x,i) = 0$.

VIII.4 If $r^{[t+1]} = r^{[t]}$, stop and set $r^* = r^{[t+1]}$; otherwise increment t by 1 and and return to Step 2.

Rate of Convergence: Theorem 8.6.6 in [68] confirms that the policy iteration algorithm converges after finitely many iterations.

Time Complexity: The tree structure of Ω_{U} allows its size to be estimated by $MQ^{\tilde{L}}$, where $\tilde{L}$ is given by Equation (3.40). In the policy evaluation step, Algorithm V is run with respect to Ω_{U}, so its complexity is $\mathcal{O}(M^3Q^{\tilde{L}})$. The complexity of the policy improvement step is $\mathcal{O}(M^2Q^{\tilde{L}})$ because the tree structure of Ω_{U} allows us to store links between $h^{[t]}(x,i)$ and $h^{[t]}(x \oplus q, j)$ directly.

Memory Complexity: The policy evaluation step has memory complexity of $\mathcal{O}(M^2Q^{\bar{L}})$ because of Algorithm V. Whereas the policy improvement step has memory requirement complexity of $\mathcal{O}(M\tilde{L}Q^{\tilde{L}})$ because x, $h^{[t]}(x,i)$, $h^{[t+1]}(x,i)$ are stored for all $(x,i) \in \Omega_{\mathrm{U}}$ in each iteration t.

Please refer to Appendix C for a numerical example to verify the complexity estimates. Note that the optimal clearing policy can also be found via a linear program of the corresponding Markov Decision Process. We modify the existing LP formulation in [68] to solve our version of stochastic clearing problem in Appendix B.

We use Algorithms VII and VIII separately to solve the following stochastic clearing problems. The results are consistent, hence assuring the correctness of the two methods.

Example 5.3.1. *Consider a stochastic clearing system with the following parameters.*

(i) The input process is the BMAP in Example 2.1.1.(iii).

(ii) The delay penalty cost is given as in Example 2.3.1.(d).

(iii) The fixed clearing cost is $K(w_t) = 5 \cdot \delta_{\{|w_t|>0\}}$.

(iv) The variable clearing cost is $C(w_t) = 0.5|w_t|$.

Result Summary	$r^*(x,1)$	$r(x,2)$
$\emptyset$	0	0
$x = [1]$	0	0
$x = [2]$	1	1
$x = [1,0]$	1	0
$x = [1,1]$	1	0
$x = [1,2]$	1	1
$x = [2,0]$	1	1
$x = [2,1]$	1	1
$x = [1,0,0]$	1	1
$x = [1,0,1]$	1	1
$x = [1,0,2]$	1	1
$x = [1,1,0]$	1	1
$x = [1,1,1]$	1	1
$x = [1,1,2]$	1	1
$x = [1,0,0,0]$	1	1
$x = [1,0,0,1]$	1	1

Table 5.2: Summary of Optimal Clearing Rule for Example 5.3.1

The optimal clearing policy and the corresponding value functions are summarized in Table 5.2. The cost functions given in Example 5.3.1 clearly satisfy Condition 5.2.1. The numerical results of this example show that the optimal stationary clearing rules depend on the underlying state i, and demonstrate the characteristics proven in Section 6.2. In the next example, we show a stochastic clearing problem with an optimal state-dependent threshold clearing policy.

Example 5.3.2. *Consider a stochastic clearing system with the following parameters.*

(i) The input process is given as the BMAP in Example 2.1.1.(iii).

(ii) The delay penalty cost is as in Example 2.3.1.(c).

(iii) The fixed clearing cost is $K(w_t) = 5 \cdot \delta_{\{|w_t|>0\}}$.

(iv) The variable clearing cost is $C(w_t) = 0$.

x	$r^*(x,1)$	$r^*(x,2)$	$H(x,2)$
$\emptyset$	0	0	0
[1]	0	0	0.5
[2]	1	0	0.5
[1,0]	1	1	2
[1,1]	1	1	2.5
[1,2]	1	1	2.5
[2,0]	1	1	2
[2,1]	1	1	2.5
[2,2]	1	1	2.5

Table 5.3: Summary of Optimal Clearing Rule in Period 1 for Example 5.3.2

The optimal clearing policy and the corresponding value functions are summarized in Table 5.3. The numerical results of Example 5.3.2 show that the optimal clearing rules can be converted into state-dependent threshold rules similar to those defined in Section 4.3. The optimal threshold policy parameters can be expressed as $\tau^*(1) = \tau^*(2) = 0.5$.

5.4 Summary of Results

Using the standard approaches for Markov decision processes (MDPs), we study the expected average cost per period problem over an infinite horizon in this chapter. Similar to the previous two chapters, we first construct an algorithm to evaluate any given clearing policy. Using the policy evaluation equations, we obtain the optimality equations and then modify the value/policy iteration algorithms to search for the optimal clearing policy parameters.

So far, we have evaluated a clearing policy only with respect to the cost objectives. However, there are many other performance measures which can be used to assess the effectiveness of a clearing policy. In the next chapter, we will use a different method

to compute the expected average cost per period, as well as to calculate several other performance measures.

Chapter 6

Expected Average Cost Model over Infinite Horizon (MAM)

In this chapter, we use the Matrix-Analytic Methods (MAM) to study the stochastic clearing problem over an infinite planning horizon, with the expected average cost per period as our objective function. MAM relies on special structures of the Markov chains of the system states to efficiently compute the stationary distribution, expect average cost, and other long-run performance measures. This approach is particularly effective in evaluating a given clearing policy. The modelling assumptions and notation used in this chapter are summarized below.

(i) The planning horizon is infinite, i.e., $T = \{1, 2, \ldots\}$.

(ii) The model is in discrete time and discrete quantities, hence the input process is modelled as a *BMAP*.

(iii) The clearing policy is given by a stationary policy $\pi = (r, r, \ldots)$, thus π and r can be used interchangeably.

(iv) The post-clearing system state process $\{(y_t, i_t), t = 1, 2, \ldots\}$ is the Markov chain of interest in this chapter.

(v) The cost functions are given by H, K, and C, which are stationary over time, but we can add the subscript t if necessary.

6.1 Policy Evaluation

Recall that in Theorem 2.2.3, we have already shown that the extended post-clearing system state space $\Phi_{(r)} \times I$ has a finite tree structure under any logical, feasible, and stationary clearing policy $\pi = (r, r, \ldots)$. After each transition, the system content y either becomes $y \oplus q$, which is one of its "children" on the tree, or it returns to the root node $\emptyset$. Therefore, we recongnize that the Markov chain $\{(y_t, i_t), t = 1, 2, \ldots\}$ is in fact a *GI/M/1* type Markov chain with tree structure. Thess types of Markov chains have been studied using Matrix-Analytic Methods, and efficient algorithms for computing the stationary distribution have been built. In this section, we modify those algorithms to help analyze stochastic clearing systems.

6.1.1 Stationary Distribution

Recall from Section 2.1.4 that, for a given stationary clearing rule r, the state space of $\{(y_t, i_t), t = 1, 2, \ldots\}$ is $\Omega_{(r)}$, which is given by Equation (2.18). However, expanding the state space into $\Phi_{(r)} \times I$ can be more efficient for computational purposes. Condition 2.2.1 guarantees that the extra dummy states are never reached if the system starts from an empty state, hence they do not interfere with our analysis of the stationary system states.

By expanding the state space of the Markov chain, we can obtain block transition probability matrices $A^{(y,r)}_{y,y\oplus q}$ and $B^{(y,r)}_{y,\emptyset}$ from Equations (2.37) to (2.39), but to simplify the

notation in this section, we shall write

$$A^{(y,r)}_{\emptyset,\emptyset} = D_0, \tag{6.1}$$

where D_0 is the block matrix of zero input in the BMAP, followed by

$$A^{(y,r)}_{y,y\oplus q} = A(y)_q \tag{6.2}$$

and

$$B^{(y,r)}_{y,\emptyset} = B(y), \tag{6.3}$$

for all $y \in \Phi_{(r)}$ and $q = 0, 1, \ldots$.

Let $\boldsymbol{\theta} = (\theta(y,i), \forall (y,i) \in \Omega_{(r)})$ be the steady-state distribution of the Markov chain $\{(y_t, i_t), t = 1, 2, \ldots\}$. Then $\boldsymbol{\theta}$ satisfies

$$\begin{cases} \theta(\emptyset, j) & = \sum_{i=1}^{M} \theta(\emptyset, i) a^{(y,r)}_{(\emptyset,i),(\emptyset,j)} + \sum_{(y,i)\in\Omega_{(r)}} \theta(y) b^{(y,r)}_{(y,i),(\emptyset,j)}, \quad \forall (\emptyset, j) \in \Omega_{(r)} \\ \theta(y \oplus q, j) & = \sum_{i=1}^{M} \theta(y,i) a^{(y,r)}_{(y,i),(y\oplus q,j)}, \quad \forall (y \oplus q, j) \in \Omega_{(r)} \end{cases} \tag{6.4}$$

After expanding the state space of the Markov chain to $\Phi_{(r)} \times I$, we can rewrite the steady-state distribution as $\boldsymbol{\theta} = (\boldsymbol{\theta}(y), \forall y \in \Phi_{(r)})$, where $\boldsymbol{\theta}(y) = (\theta(y,1), \theta(y,2), \ldots, \theta(y,M))$. Using the matrix notation introduced in Equations (2.38) and (2.39), we can rewrite Equation (6.4) as

$$\begin{cases} \boldsymbol{\theta}(\emptyset) & = \boldsymbol{\theta}(\emptyset) D_0 + \sum_{y\in\Phi_{(r)}} \boldsymbol{\theta}(y) B(y) \\ \boldsymbol{\theta}(y \oplus q) & = \boldsymbol{\theta}(y) A(y)_q, \quad \forall y \oplus q \in \Phi_{(r)} \end{cases} \tag{6.5}$$

For any pair of sequences $x = \left[x_{[l]}, \ldots, x_{[1]}\right]$ and $y = \left[y_{[n]}, \ldots, y_{[1]}\right]$, such that $x \in \Phi_{(r)}$

and $x \oplus y \in \Phi_{(r)}$, let the *n-step transition probabilities without clearing* be

$$r_{(x,i),(x\oplus y,j)} = \sum_{j_1 \in I} \cdots \sum_{j_{n-1} \in I} a^{(y,r)}_{(x,i),(x\oplus y_L(1),j_1)} a^{(y,r)}_{(x\oplus y_L(1),j_1),(x\oplus y_L(2),j_2)} \cdots a^{(y,r)}_{(x\oplus y_L(n-1),j_{n-1}),(x\oplus y,j)} \tag{6.6}$$

where $y_L(n)$ is the left sub-sequence operator from Definition 2.1.1. In the matrix form, we have

$$\tilde{R}_{\emptyset,\emptyset} = (I - D_0)^{-1}, \tag{6.7}$$

since $\emptyset \oplus 0 = \emptyset$, and

$$\tilde{R}_{x,y} = A(x)_{y_{[n]}} A(x \oplus y_L(1))_{y_{[n-1]}} \ldots A(x \oplus y_L(n-1))_{y_{[1]}}, \tag{6.8}$$

for all $x \in \Phi_{(r)}$ and $y \in \Phi_{(r)}$ but $y \neq \emptyset$.

In any arbitrary clearing cycle, the system content can only be y at most once, except when $y = \emptyset$. Therefore, entry (i, j) in matrix $\tilde{R}_{\emptyset,\emptyset}\tilde{R}_{\emptyset,y}$ is in fact the expected time spent in state (y, j) during the cycle, given that the cycle started from state $(\emptyset, i)$, for $i, j \in I$ and $y \in \Phi_{(r)}$. These matrices are similar to the rate matrix R for the $GI/M/1$ type Markov chains (see [37] and [60] for more details on the rate matrix R).

Let us introduce a more convenient notation for later use, namely

$$\begin{cases} R(\emptyset) = I \\ R(y) = \tilde{R}_{\emptyset,y}, \forall y \in \Phi_{(r)} \text{ and } y \neq \emptyset. \end{cases} \tag{6.9}$$

Based on Equations (2.34), (2.35), (6.4), (6.6), and the definition of stationary distribution of a Markov chain, the following results can be obtained.

Theorem 6.1.1. *Under the assumptions given in Theorem 2.2.3, the stationary distribution of the Markov chain* $\{(y_t, i_t), t = 1, 2, \ldots\}$, *subjected to the stationary policy*

$\pi = (r, r, \ldots)$, *can be expressed as*

$$\theta(y,j) = \sum_{i=1}^{M} \theta(\emptyset, i) r_{(\emptyset,i),(y,j)}, \forall (y,j) \in \Omega_{(r)} \text{ and } (y,j) \neq (\emptyset, j), \tag{6.10}$$

where $\theta(\emptyset, j)$ *is the unique solution to the linear system*

$$\begin{cases} \theta(\emptyset,j) = \theta(\emptyset,i) \left(a^{(y,r)}_{(\emptyset,i),(\emptyset,j)} + b^{(y,r)}_{(\emptyset,i),(\emptyset,j)} + \sum\limits_{(y,m)\in\Omega_{(r)}:(y,m)\neq(\emptyset,j)} r_{(\emptyset,i),(y,m)} b^{(y,r)}_{(y,m),(\emptyset,j)} \right), \forall j \in I \\ 1 = \sum\limits_{j=1}^{M} \theta(\emptyset,j) \left(1 + \sum\limits_{(y,m)\in\Omega_{(r)}:(y,m)\neq(\emptyset,j)} r_{(\emptyset,i),(y,m)} \right) \end{cases} \tag{6.11}$$

Proof. Let $P^{(x,r)}$ be the transition probability matrix for the Markov chain $\{(y_t, i_t), t = 1, 2, \ldots\}$. The components of $P^{(x,r)}$ are given in Equation (2.33). It can be easily verified that, according to Equation (6.4), $\boldsymbol{\theta}$ is a solution to the linear system $\boldsymbol{\theta} P^{(x,r)} = \boldsymbol{\theta}$ and $\boldsymbol{\theta e} = 1$, which can be simplified into Equations (6.10) and (6.11). The uniqueness of the solution is guaranteed by Theorem 2.2.3. □

The following corollary follows directly from Theorem 6.1.1 by expanding the state space and using the matrix notation defined earlier.

Corollary 6.1.2. *Under the assumptions given in Theorem 2.2.3, the stationary distribution of the Markov chain* $\{(y_t, i_t), t = 1, 2, \ldots\}$, *subjected to the stationary policy* $\pi = (r, r, \ldots)$, *can be expressed as*

$$\boldsymbol{\theta}(y) = \boldsymbol{\theta}(\emptyset) R(y), \forall y \in \Phi_{(r)}, \tag{6.12}$$

where $\boldsymbol{\theta}(\emptyset)$ *satisfies the linear system*

$$\begin{cases} \boldsymbol{\theta}(\emptyset) = \boldsymbol{\theta}(\emptyset) \left(D_0 + \sum_{y \in \Phi_{(r)}} R(y) B(y) \right) \\ 1 = \boldsymbol{\theta}(\emptyset) \sum_{y \in \Phi_{(r)}} R(y) \boldsymbol{e} \end{cases}, \tag{6.13}$$

if such a stationary distribution exists.

Since the input process is not affected by the shipment consolidation process, $\boldsymbol{\theta}$ is directly related to $\boldsymbol{\theta}_{\mathrm{a}}$, which is the stationary distribution of the underlying Markov chain of the input process according to Section 2.1.2.

Proposition 6.1.3. *Under the assumptions stated in Theorem 2.2.3, we have*

$$\sum_{y \in \Phi_{(r)}} \boldsymbol{\theta}(y) = \boldsymbol{\theta}_{\mathrm{a}}. \tag{6.14}$$

Proof. We begin the proof by showing that $\sum_{y \in \Phi_{(r)}} \boldsymbol{\theta}(y) D = \sum_{y \in \Phi_{(r)}} \boldsymbol{\theta}(y)$. The left hand

side of the equation can be evaluated as

$$
\begin{aligned}
&\sum_{y\in\Phi_{(r)}} \boldsymbol{\theta}(y)D \\
&= \sum_{y\in\Phi_{(r)}} \boldsymbol{\theta}(y)(\sum_{q=0}^{Q} A(y)_q + B(y)) \\
&= \sum_{y\in\Phi_{(r)}} \boldsymbol{\theta}(\emptyset)R(y)(\sum_{q=0}^{Q} A(y)_q + B(y)) \\
&= \boldsymbol{\theta}(\emptyset)[\sum_{q=0}^{Q} A(\emptyset)_q + B(\emptyset)] + \boldsymbol{\theta}(\emptyset) \sum_{y\in\Phi_{(r)}:y\neq\emptyset} R(y)[\sum_{q=0}^{Q} A(y)_q + B(y)] \\
&= \boldsymbol{\theta}(\emptyset) \sum_{q=1}^{Q} A(\emptyset)_q + \sum_{y\in\Phi_{(r)}:y\neq\emptyset} \boldsymbol{\theta}(y) \sum_{q=0}^{Q} A(y)_q + \boldsymbol{\theta}(\emptyset) \\
&= \sum_{y\oplus q\in\Phi_{(r)}:y\oplus q\neq\emptyset} \boldsymbol{\theta}(y\oplus q) + \boldsymbol{\theta}(\emptyset) \\
&= \sum_{y\in\Phi_{(r)}} \boldsymbol{\theta}(y).
\end{aligned}
$$

The first equality follows from Equation (2.41), the second uses Equation (6.12), the forth equality is a result of Equation (6.13), and finally, the fifth can be seen from the one-step state transitions without clearing. Since $\boldsymbol{\theta}_{\mathrm{a}}$ is the unique stationary distribution of the underlying Markov chain D and $\left(\sum_{y\in\Phi_{(r)}} \boldsymbol{\theta}(y)\right)\boldsymbol{e} = 1$, we must have $\sum_{y\in\Phi_{(r)}} \boldsymbol{\theta}(y) = \boldsymbol{\theta}_{\mathrm{a}}$. This completes the proof. □

Now let us construct an algorithm to compute the stationary distribution for $\{(y_t, i_t), t = 1, 2, \ldots\}$ under a given policy. Since $\Phi_{(r)}$ has a tree structure, we can modify BFT Procedure II introduced in Section 2.1.4 to help us construct and navigate the state space of the Markov chain.

Stationary Distribution Procedure

Step 1 Initialize a list V with $\{\emptyset, I, D - D_0\}$ as the first entry; set counter $n = 1$ to

record the number of entries in V; and denote the three components in the n-th entry in V as $V(n,1)$, $V(n,2)$, and $V(n,3)$.

Step 2 Initialize a list U; for each $q = 1, \ldots, Q$, compute $A(\emptyset)_q$ by Equation (2.38) and store $\{n, q, A(\emptyset)_q\}$ as separate entries in U.

Step 3 If U is empty, proceed to Step 5; otherwise, read and delete the next entry from U according to the First-In-First-Out (FIFO) rule, store the three components from the entry as n_{temp}, q_{temp}, and A_{temp}; and let $x_{\text{temp}} = V(n_{\text{temp}}, 1) \oplus q_{\text{temp}}$.

Step 4 If $A_{\text{temp}} \neq 0$, store $\{x_{\text{temp}}, V(n_{\text{temp}}, 2) * A_{\text{temp}}, D\}$ as a new entry in V and increment n by one; update $V(n_{\text{temp}}, 3) = V(n_{\text{temp}}, 3) - A_{\text{temp}}$; and for each $q = 1, \ldots, Q$, compute $A(x_{\text{temp}})_q$ by Equation (2.38) and store $\{n, q, A(x_{\text{temp}})_q\}$ as new entries in U; then go back to Step 3.

Step 5 Solve the system of equations given in (6.13), where $\Phi_{(r)} = \{V(j,1), j = 1, 2, \ldots, n\}$, $\{R(y), y \in \Phi_{(r)}\} = \{V(j,2), j = 1, 2, \ldots, n\}$, and $\{B(y), y \in \Phi_{(r)}\} = \{V(j,3), j = 1, 2, \ldots, n\}$; output the result as $\boldsymbol{\theta}$.

Time Complexity: Since $\Phi_{(r)}$ has a tree structure, we can see that its size is limited by $Q^{\bar{L}_{(r)}}$, where Q is the maximum input size and $\bar{L}$ is the maximum input age allowed. For each $y \in \Phi_{(r)}$, the total computation time for matrices $A(y)_q$, $B(y)$, and $R(y)$ scales by $\mathcal{O}(QM)$. Therefore, it is easy to conclude that the time complexity of the Stationary Distribution Procedure is $\mathcal{O}(MQ^{\bar{L}_{(r)}})$.

Memory Complexity: The amount of memory required scales by $\mathcal{O}(M^2 Q^{\bar{L}_{(r)}})$ because we need to store the $M \times M$ matrices $A(y)_q$, $B(y)$, and $R(y)$, for all $y \in \Phi_{(r)}$ and $q = 0, 1, \ldots, Q$.

6.1.2 Clearing Cycles

If we assume that a *clearing cycle* starts from an empty system and ends with the next clearing event, the following proposition follows from renewal theory.

Lemma 6.1.4. *The underlying state at the start of each clearing cycle is itself a Markov chain. Let P_c and $\boldsymbol{\theta}_c$ denote the corresponding transition probability matrix and stationary distribution, respectively. Then we have*

$$P_c = (I - D_0)^{-1} \left(\sum_{y \in \Phi_{(r)}} R(y)B(y) \right), \tag{6.15}$$

where

$$\boldsymbol{\theta}_c = \boldsymbol{\theta}_c P_c \text{ and } \boldsymbol{\theta}_c \boldsymbol{e} = 1. \tag{6.16}$$

Proof. According to Equations(6.7) and (6.8)

$$P_c = \tilde{R}_{\emptyset,\emptyset} \left(\sum_{y \in \Phi_{(r)}: y \neq \emptyset} \tilde{R}_{\emptyset,y} B(y) \right) = (I - D_0)^{-1} \left(\sum_{y \in \Phi_{(r)}} R(y)B(y) \right),$$

which are the transition probabilities of a single cycle. □

Note: Corrolary 6.1.2 requires the existence of the stationary distribution $(\boldsymbol{\theta}, \forall y \in \Phi_{(r)})$. However, due to the dummy states in $\Phi_{(r)} \times I$, the corresponding Markov chain may not be irreducible, so we cannot verify if $\boldsymbol{\theta}$ exists. However, $\boldsymbol{\theta}_c$ always exists because the underlying Markov chain is irreducible. In fact, the following lemma will show that $\boldsymbol{\theta}_c$ gives us an alternative method for computing $\boldsymbol{\theta}$.

Let p_c be the probability of clearing during an arbitrary period in the long run.

Lemma 6.1.5. *Under the assumptions given in Theorem 2.2.3, we have*

$$p_c = \boldsymbol{\theta}(\emptyset)(I - D_0)\boldsymbol{e}. \tag{6.17}$$

and

$$\boldsymbol{\theta}_{\mathrm{c}} = \frac{\boldsymbol{\theta}(\emptyset)\left(\sum_{y\in\Phi_{(r)}} R(y)B(y)\right)}{p_{\mathrm{c}}} = \frac{\boldsymbol{\theta}(\emptyset)(I - D_0)}{p_{\mathrm{c}}}. \tag{6.18}$$

Proof. Recall from Section 2.2.2 that there are two types of transitions into the empty system state: (i) system remains empty if no input is received, i.e., $\emptyset \oplus 0 \to \emptyset$; or (ii) system is cleared, i.e., $y \oplus q \to \emptyset$. Therefore, $\boldsymbol{\theta}_{\mathrm{c}}$ corresponds to the probabilities of the second type of transition, but conditioned on the event that the system has just been cleared. By Equations (2.39) and (6.13), the probability of clearing can be expressed as

$$p_{\mathrm{c}} = \sum_{y\in\Phi_{(r)}} \boldsymbol{\theta}(y)B(y)\boldsymbol{e} = \boldsymbol{\theta}(\emptyset)(I - D_0)\boldsymbol{e}.$$

This completes the proof. □

6.1.3 Long-Run Performance Measures

The effectiveness of a clearing policy is not only measured by the relevant costs. In our research, we are interested in the following performance measures as well.

L_{c}: the length of a *clearing cycle*, i.e., the time between two consecutive clearings.

L_{idle}: the *inactive time* in a clearing cycle, i.e., the number of consecutive periods with no inputs right after the previous clearing.

L_{active}: the time in a clearing cycle that the system is actively accumulating inputs.

W_{c}: the total cumulative quantities upon clearing.

N_{c}: the number of non-zero inputs upon clearing.

$A_{\mathrm{c}}^{\mathrm{TOT}}$: the total age of non-zero inputs upon clearing.

$A_{\mathrm{c}}^{\mathrm{AVG}}$: the average age of non-zero inputs upon clearing.

Note that all of the above performance measures are to be computed as long-run averages. We shall demonstrate immediately that these measures are independent of the initial system state.

For infinite horizon problems, if no partial clearing is allowed, we can use the time of clearing to divide the process into clearing cycles. In this way, the initial clearing cycle begins with a given state (x, i), but all subsequent cycles begin from the empty state immediately after a clearing. Recall from Theorem 2.2.3 that, if a given clearing policy is logical and feasible, then the Markov chain $\{(y_t, i_t), t = 1, 2, \ldots\}$ is positive recurrent. This means the expected time to return to the empty state is finite, regardless of the initial state. Over an infinite horizon, the effect of the first cycle is negligible if we are to take averages over all cycles. Thus, we can ignore the first cycle and assume that all clearing cycles start with an empty state.

The following proposition suggests that the repeated clearing cycles form a renewal process with i.i.d. cycle lengths.

Proposition 6.1.6. *(Theorem 3.4 in [12]) Under the assumptions given in Theorem 2.2.3, L_{idle} has a discrete phase-type distribution with PH-representation $(\boldsymbol{\theta}_{\text{c}}, D_0)$; and L_{c} has a discrete phase-type distribution with PH-representation $((\boldsymbol{\theta}_c, 0, ..., 0), \tilde{P}^{(y,r)})$, where $\tilde{P}^{(y,r)}$ is obtained by removing blocks $B(y), \forall y \in \Phi_{(r)}$, from the transition probability matrix $P^{(y,r)}$.*

The distributions of L_{c} and L_{idle} can be given explicitly as follows:

$$\begin{aligned} P\{L_{\text{c}} = n\} &= \boldsymbol{\theta}_{\text{c}} \left(\sum_{j=0}^{n-1} D_0^j \sum_{y:\ y\in\Phi_{(r)},\ \mathcal{L}(y)=n-j} R(y)B(y) \right) \boldsymbol{e}, \text{ for } n = 1, 2, ... \\ P\{L_{\text{idle}=n}\} &= \boldsymbol{\theta}_{\text{c}} D_0^n (I - D_0)\boldsymbol{e}, \text{ for } n = 0, 1, 2, ... \end{aligned} \tag{6.19}$$

Simple expressions can be obtained for the expected values of $L_{\rm c}$ and $L_{\rm idle}$:

$$\begin{aligned} E[L_{\rm c}] &= \frac{1}{p_{\rm c}} = \frac{1}{\boldsymbol{\theta}(\emptyset)(I - D_0)\boldsymbol{e}} \\ E[L_{\rm idle}] &= \frac{\boldsymbol{\theta}(\emptyset)\boldsymbol{e}}{p_{\rm c}} \end{aligned} \tag{6.20}$$

The expected value of $L_{\rm active}$ is $E[L_{\rm active}] = E[L_{\rm c}] - E[L_{\rm idle}]$.

Let W be the total accumulated weight in the system at the beginning of an arbitrary period. Under the assumptions in Theorem 2.2.3, the distribution of W and the mean of W can be expressed in terms of $\boldsymbol{\theta}(\emptyset)$ and $\{R(y),\ y \in \Phi_{(r)}\}$ as follows:

$$\begin{aligned} P\{W = w\} &= \boldsymbol{\theta}(\emptyset) \left(\sum_{y \in \Phi_{(r)}:\ |y| = w} R(y) \right) \boldsymbol{e}, \text{ for } w = 0, 1, ..., \bar{Q}; \\ E[W] &= \boldsymbol{\theta}(\emptyset) \left(\sum_{y \in \Phi_{(r)}} |y| R(y) \right) \boldsymbol{e}. \end{aligned} \tag{6.21}$$

Recall that $W_{\rm c}$, $N_{\rm c}$, $A_{\rm c}^{\rm TOT}$, and $A_{\rm c}^{\rm AVG}$ are, respectively, the total cumulative quantity, the number of non-zero inputs, the total age of non-zero inputs, and the average age of non-zero inputs upon clearing. Under the assumptions in Theorem 2.2.3, their distributions can be obtained by conditioning on the event that the system has just been cleared in the period.

Before constructing the distribution functions of these performance measures, let us first define

$$\mathcal{N}(y) = \sum_{j=1}^{l} \delta_{\{y_{[j]} > 0\}} \tag{6.22}$$

to compute the number of non-zero entries in sequence y, and another function

$$\mathcal{A}(y) = \sum_{j=1}^{l} j \delta_{\{y_{[j]} > 0\}} \tag{6.23}$$

to find the total age of the inputs in that sequence y.

Now we can write the distribution functions of W_c, N_c, A_c^{TOT}, and A_c^{AVG} as follows.

$$P\{W_c = w\} = \frac{\boldsymbol{\theta}(\emptyset)}{p_c} \sum_{y \in \Phi_{(r)}} R(y) \left(\sum_{q=0:\ |y|+q=w}^{Q} D_q - A(y)_q \right) \boldsymbol{e}, \tag{6.24}$$

for $w = 1, 2, ..., \bar{Q} + Q$.

$$P\{N_c = n\} = \frac{\boldsymbol{\theta}(\emptyset)}{p_c} \sum_{y \in \Phi_{(r)}} R(y) \left(\sum_{q=0:\ \mathcal{N}(y \oplus q)=n}^{Q} D_q - A(y)_q \right) \boldsymbol{e}, \tag{6.25}$$

for $n = 1, 2, ..., \bar{L} + 1$.

$$P\{A_c^{TOT} = l\} = \frac{\boldsymbol{\theta}(\emptyset)}{p_c} \sum_{y \in \Phi_{(r)}} R(y) \left(\sum_{q=0:\ \mathcal{A}(y \oplus q)=l}^{Q} D_q - A(y)_q \right) \boldsymbol{e}, \tag{6.26}$$

for $l = 0, 1, \ldots, \bar{L}$.

$$P\{A_c^{AVG} \leq l\} = \frac{\boldsymbol{\theta}(\emptyset)}{p_c} \sum_{y \in \Phi_{(r)}} R(y) \left(\sum_{q=0:\ \frac{\mathcal{A}(y \oplus q)}{\mathcal{N}(y \oplus q)} \leq l}^{Q} D_q - A(y)_q \right) \boldsymbol{e}, \tag{6.27}$$

for $0 \leq l \leq \bar{L}$.

The distributions of A_c^{TOT} and A_c^{AVG} can be used to determine the service level of the system in the long run (i.e., the probability of having undesirable average delay, or the expected average delay per input). The means of W_c, N_c, A_D^{TOT}, and A_D^{AVG} can be computed by the following equations.

$$E[W_c] = \frac{\boldsymbol{\theta}(\emptyset)}{p_c} \sum_{y \in \Phi_{(r)}} R(y) \left(\sum_{q=0}^{Q} (|y| + q)(D_q - A(y)_q) \right) \boldsymbol{e} \tag{6.28}$$

$$E[N_c] = \frac{\boldsymbol{\theta}(\emptyset)}{p_c} \sum_{y \in \Phi_{(r)}} R(y) \left(\sum_{q=0}^{Q} \mathcal{N}(y \oplus q)(D_q - A(y)_q) \right) \boldsymbol{e} \tag{6.29}$$

$$E[A_{\mathrm{c}}^{\mathrm{TOT}}] = \frac{\boldsymbol{\theta}(\emptyset)}{p_{\mathrm{c}}} \sum_{y \in \Phi_{(r)}} R(y) \left(\sum_{q=0}^{Q} \mathcal{A}(y \oplus q)(D_q - A(y)_q) \right) \boldsymbol{e} \tag{6.30}$$

$$E[A_{\mathrm{c}}^{\mathrm{AVG}}] = \frac{\boldsymbol{\theta}(\emptyset)}{p_{\mathrm{c}}} \sum_{y \in \Phi_{(r)}} R(y) \left(\sum_{q=0}^{Q} \frac{\mathcal{A}(y \oplus q)}{\mathcal{N}(y \oplus q)}(D_q - A(y)_q) \right) \boldsymbol{e} \tag{6.31}$$

Similar to Proposition 6.1.3, the clearing process has no effect on the long-run quantity-receiving rate and input-receiving rate, thus leading to the following results.

Proposition 6.1.7. *For a given clearing policy, under the assumptions given in Theorem 2.2.3, we have*

$$E[W_{\mathrm{c}}] = \lambda_{\mathrm{q,a}} E[L_{\mathrm{c}}] \tag{6.32}$$

and

$$E[N_{\mathrm{c}}] = \lambda_{\mathrm{o,a}} E[L_{\mathrm{c}}]. \tag{6.33}$$

Proof. We begin with the first equation, for which $\lambda_{\mathrm{q,a}} = \boldsymbol{\theta}_{\mathrm{a}}(\sum_{q=0}^{Q} q D_q)\boldsymbol{e}$, $E[L_{\mathrm{c}}] = 1/p_{\mathrm{c}}$, and $E[W_{\mathrm{c}}]$ is given in Equation (6.28). Thus we find

$$\begin{aligned}
\frac{E[W_{\mathrm{c}}]}{E[L_{\mathrm{c}}]} &= \boldsymbol{\theta}(\emptyset) \sum_{y \in \Phi_{(r)}} R(y) \left(\sum_{q=0}^{Q} (|y| + q)(D_q - A(y)_q) \right) \boldsymbol{e} \\
&= \sum_{y \in \Phi_{(r)}} \boldsymbol{\theta}(y) \left(\sum_{q=0}^{Q} (|y| + q)(D_q - A(y)_q) \right) \boldsymbol{e} \\
&= \sum_{y \in \Phi_{(r)}} \boldsymbol{\theta}(y) \left(\sum_{q=0}^{Q} q D_q \boldsymbol{e} + |y| D\boldsymbol{e} - \sum_{q=0}^{Q} |y \oplus q| A(y)_q \boldsymbol{e} \right) \\
&= \lambda_{\mathrm{q,a}} + \sum_{y \in \Phi_{(r)}} |y| \boldsymbol{\theta}(y) D\boldsymbol{e} - \sum_{y \in \Phi_{(r)}} \sum_{q=0}^{Q} |y \oplus q| \boldsymbol{\theta}(y) A(y)_q \boldsymbol{e} \\
&= \lambda_{\mathrm{q,a}},
\end{aligned}$$

since $D\boldsymbol{e} = \boldsymbol{e}$ and $\boldsymbol{\theta}(y)A(y)_q = \boldsymbol{\theta}(y \oplus q), \forall y \oplus q \in \Phi_{(r)}$. The proof for the second equation is similar to the first. We just need to substitute $\mathcal{N}(y)$ for $|y|$. □

Proposition 6.1.7 is quite intuitive because all quantities and inputs received must eventually be cleared. The two equations can also be used to check the accuracy of the stationary distribution computed.

6.1.4 Expected Average Cost per Period

Now we are ready to compute the expected average cost per period over an infinite horizon for a given clearing policy. Note that the clearing costs are only charged once per cycle as $K(W_c)$ and $C(W_c)$, where W_c is the total cumulative quantity upon clearing. It is easy to see that, because of Proposition 6.1.6 and Equation (6.24), the cleared quantities in all clearing cycles except the first one are i.i.d.. Therefore, renewal reward theory allows us to compute the expected average clearing costs per period over an infinite horizon as

$$\frac{E[K(W_c)]}{E[L_c]} = \hat{k}\boldsymbol{\theta}(\emptyset)(I - D_0)\boldsymbol{e} \tag{6.34}$$

and

$$\frac{E[C(W_c)]}{E[L_c]} = \boldsymbol{\theta}(\emptyset) \sum_{y \in \Phi_{(r)}} R(y) \sum_{q=0}^{Q} C(|y| + q)(D_q - A(y)_q)\boldsymbol{e}. \tag{6.35}$$

In contrast to the clearing costs, the delay penalty cost is charged every period to the post-clearing system content. We can safely assume that when the system is empty, no penalty is incurred. Therefore, we use $\boldsymbol{\theta}$ to compute the expected average delay penalty cost per period over an infinite horizon, which is given by

$$\mathcal{C}_H^{r,A} = \boldsymbol{\theta}(\emptyset) \sum_{y \in \Phi_{(r)}} R(y)H(y)\boldsymbol{e}. \tag{6.36}$$

Adding the three types of costs together, we get

$$\mathcal{C}_{<1,\infty>}^{r,A}(x,i) = \lim_{N \to \infty} \mathcal{C}_{<1,N>}^{r,A}(x,i) = \frac{E[K(W_c)] + E[C(W_c)]}{E[L_c]} + \mathcal{C}_H^{r,A}. \tag{6.37}$$

The following algorithm summarizes the key steps in evaluating the long-run performance measures and average costs of a given clearing policy.

Algorithm IX: Stationary Policy Evaluation Algorithm

IX.1 Construct $\Phi_{(r)}$, $\{R(x), x \in \Phi_{(r)}\}$, and $\{B(x), x \in \Phi_{(r)}\}$, then compute the stationary distribution $\boldsymbol{\theta}$ through the Stationary Distribution Procedure.

IX.2 For all $y \in \Omega_{(r)}$, compute and store $\mathcal{N}(y)$ and $\mathcal{A}(y)$ according to Equations (6.22) and (6.23), respectively.

IX.3 Calculate the distributions of various long-run performance measures using Equations (6.17), (6.19), (6.21), and (6.24) to (6.27). (This step is optional since the full distributions are not always required)

IX.4 Calculate the means of various long-run performance measures using Equations (6.20), (6.21), and (6.28) to (6.31).

IX.5 Calculate the long-run average costs according to Equations (6.34) to (6.37).

Time Complexity: Step IX.1 is essentially the Stationary Distribution Procedure, which has the time complexity of $\mathcal{O}(MQ^{\bar{L}_{(r)}})$. Step IX.2 has the time complexity of $\mathcal{O}(\bar{L}_{(r)}Q^{\bar{L}_{(r)}})$ since for each $y \in \Phi_{(r)}$, computing the functions $\mathcal{N}(y)$ and $\mathcal{A}(y)$ requires checking each entry in the sequence. Computing the entire distributions in Step IX.3 is not required. However, for a particular value, the density functions and distribution functions can be computed in $\mathcal{O}(Q^{\bar{L}_{(r)}})$ time. Lastly, Steps IX.4 and IX.5 both have time complexities of $\mathcal{O}(Q^{\bar{L}_{(r)}})$.

Compared to Algorithm V in Section 5.1.2, Algorithm IX has simpler time complexity because it takes advantage of the *GI/M/1* tree structure. This is the main advantage of

using MAM to evaluate a given clearing policy, especially when the number of underlying states is large.

Memory Requirement Complexity: The bulk of the memory is required for Step IX.1. This is the Stationary Distribution Procedure, and thus has the complexity of $\mathcal{O}(M^2 Q^{\bar{L}_{(r)}})$.

We demonstrate Algorithm IX through the following numerical example.

Example 6.1.1. *Consider nine different stochastic clearing problems with the following parameters.*

(i) Three different input processes described in Example 2.1.1.

(ii) Three different delay penalty cost functions H described in Example 2.3.1.(ii) to (iv).

(iii) The fixed clearing cost is given as $K(w_t) = 10 \cdot \delta_{\{|w_t|>0\}}$.

(iv) The variable clearing cost is given as $C(w_t) = 0.5|w_t|$.

(v) The clearing policy is given as

$$r(x_t, i_t) = \begin{cases} 0, & \text{if } |x_t| < 5 \text{ and } \mathcal{L}(x_t) < 4 \\ 1, & \text{otherwise} \end{cases}.$$

The expected average costs for the different problems are computed by Algorithm IX and summarized in the following table.

$H \backslash BMAP$	Example 2.1.1.(i)	Example 2.1.1.(ii)	Example 2.1.1.(iii)
Example 2.3.1.(b)	1.8909	1.3964	1.6622
Example 2.3.1.(c)	1.8954	1.4051	1.6328
Example 2.3.1.(d)	1.5471	1.1507	1.4898

Table 6.1: Summary of Expected Average Costs for Example 6.1.1

6.2 Computing the Optimal Policies

The general properties of the optimal clearing policy shown in Section 5.2 continue to hold in the MAM approach. In this section, we identify a couple of additional properties and then look at a special case of the problem.

6.2.1 General Properties

Recall that each downward path on the tree structure of Φ represents a sample path of the input clearing process. Assuming that the process always starts with an empty system, any stationary clearing rule r will determine the point where each sample path terminates and the process goes back to the root node $\emptyset$. Thus, the policy can be translated into a set of "cut-off" points on all sample paths of Φ for each underlying state i. In other words, the clearing rule r can be mapped onto M different sub-trees of Φ, where M is the number of underlying states. Each sub-tree contains the system contents that do not require clearing while the system is in underlying state i.

Therefore, we can determine the optimal clearing policy by traversing through Φ and finding the sets of cut-off points which minimize our objective function altogether. Of course, the key is how to execute the search process. To design such a process to try to find the optimal clearing policy, we start by showing some general properties of the optimal clearing rule r^*.

First, let us make an observation that the optimal clearing rule r^* is not always identical for different underlying state $i \in I$, i.e., r^* is a function of both the system content and the underlying states. This is because the Markovian input process actually depends on the underlying state of the system, so for the same system content but distinct underlying states, the optimal clearing decision may differ because it must also take the future input quantities iinto consideration.

For example, if the current underlying state signals possible large input quantities in the immediate future, then it might be better to hold on to the outstanding inputs for a little while and combine them with the large future inputs. On the other hand, if the underlying state suggests low input probabilities for a while, it might be better to clear the system to avoid prolonged delay of outstanding inputs. The nature of a Markovian input process usually implies that the optimal clearing policy depends on the underlying state at the beginning of each period.

Based on Lemma 5.2.1, we see that the optimal sub-trees have finite sizes. This is because as we proceed down an arbitrary sample path of Φ, there are only finitely many sequences $x \in \Phi$ such that $H(x) \leq \hat{k}$ according to Condition 5.2.1.(v).

Next, we show that the objective cost function can be simplified by removing cost items that are not affected by the policy.

Lemma 6.2.1. *Under Condition 5.2.1, the expected average variable clearing cost per period does not depend on the clearing policy.*

Proof. According to Equation (6.35), the expected variable clearing cost per period can be computed as $\frac{E[C(W_c)]}{E[L_c]}$. Since that variable clearing cost is assumed to be linear in the cleared quantities by Condition 5.2.1.(ii), we have

$$\frac{E[C(W_c)]}{E[L_c]} = \frac{C(E[W_c])}{E[L_c]} = \frac{\hat{c}E[W_c]}{E[L_c]}.$$

Proposition 6.1.7 suggests that

$$\frac{E[W_c]}{E[L_c]} = \lambda_{q,a}.$$

Therefore, $\frac{E[C(W_c)]}{E[L_c]}$ is independent of the clearing policy. □

Lemma 6.2.1 essentially allows us to ignore the variable clearing cost when we are trying to determine the optimal clearing policy. The next lemma indicates an upper bound for the optimal expected average total cost per period.

Lemma 6.2.2. *Under Condition 5.2.1, the minimal expected average total cost per period is less than or equal to the fixed clearing cost, i.e.,*

$$\mathcal{C}^{r^*,\mathrm{A}}_{<1,\infty>}(x,i) \leq \hat{k}. \tag{6.38}$$

Proof. If we clear the system every period unless it is empty, then no delay penalty is ever incurred. Since we can ignore the variable clearing cost, the expected average total cost can be computed as

$$\frac{E[K(W_{\mathrm{c}})]}{E[L_{\mathrm{c}}]} = \frac{\hat{k}E[N_{\mathrm{c}}]}{E[L_{\mathrm{c}}]} = \lambda_{\mathrm{o,a}}\hat{k},$$

by Proposition 6.1.7. This cost is less than $\hat{k}$ because we assume that there will be no more than one input per period. The desired result is obtained since the optimal policy r^* cannot do worse than that. □

6.2.2 State-Independent Threshold Policies

In this section, first define a "state-independent threshold policy" that is similar to the policy in Definition 4.3.1. We then prove that the optimal clearing policy can be determined by a delay penalty threshold if there is only a single underlying state, i.e., $M = 1$ and the input process is a compound renewal process.

Definition 6.2.1. *For given $\tau > 0$, clearing policy $\pi = (r^\tau, r^\tau, \ldots)$ is defined as*

$$r^\tau(x,i) = \begin{cases} 0, & \text{if } H(x) \leq \tau \\ 1, & \text{otherwise} \end{cases}, \tag{6.39}$$

for all $(x,i) \in \Omega$.

Note that unlike Definition 4.3.1, τ here is a single threshold that is applied across all underlying states.

We shall denote the corresponding expected average total cost per period as $\mathcal{C}^{\tau,\mathrm{A}}_{<1,\infty>}$, and let τ^* be the delay-penalty threshold that minimizes the expected average total cost per period. To simplify our notation in the remainder of this section, we write

$$\mathcal{C}^{r,\mathrm{A}}_{<1,\infty>} = \mathcal{C}(r),$$

for an arbitrary stationary clearing rule r, and

$$\mathcal{C}^{\tau,\mathrm{A}}_{<1,\infty>} = \mathcal{C}(\tau).$$

for any delay-penalty threshold policy with parameter τ. We can separate the expected average total cost per period into two components

$$\mathcal{C}^{r,\mathrm{A}}_{<1,\infty>} = \mathcal{C}_{\mathrm{K}}(r) + \mathcal{C}_{\mathrm{H}}(r),$$

where the first component is the expected clearing cost per period and the second component is the expected delay penalty cost per period.

Lemma 6.2.3. *Under Condition 5.2.1, if $\tau = \hat{k}$, then $\mathcal{C}(\tau^*) \leq \mathcal{C}(\tau) \leq \hat{k}$.*

Proof. Assume that at the beginning of each clearing cycle, there is no accumulated input; the clearing cost $\hat{k}$ can be allocated to this period, and no cost is charged during the subsequent inactive periods. By the definition of the delay-penalty threshold policy and Lemma 5.2.1, the cost incurred in each subsequent active clearing period is less than or equal to $\hat{k}$. Then the cost incurred in each period is always capped by $\hat{k}$, which leads to the desired result. □

It is intuitive that, as we raise τ, the delay penalty increases but the clearing cost decreases. The optimal threshold will strive to balance the two types of costs. Therefore,

it is reasonable to conjecture that $\mathcal{C}(\tau)$ is unimodal in τ for $\tau \geq 0$. The following numerical examples support the conjecture.

Example 6.2.1. *Suppose that* $\hat{k} = 15$ *and the delay penalty function is given as in Example 2.3.1.(d). For the three different input processes of Example 2.1.1, we plot* $\mathcal{C}(\tau)$ *over* τ *for* $0 \leq \tau \leq \hat{k}$ *in Figure 6.1.*

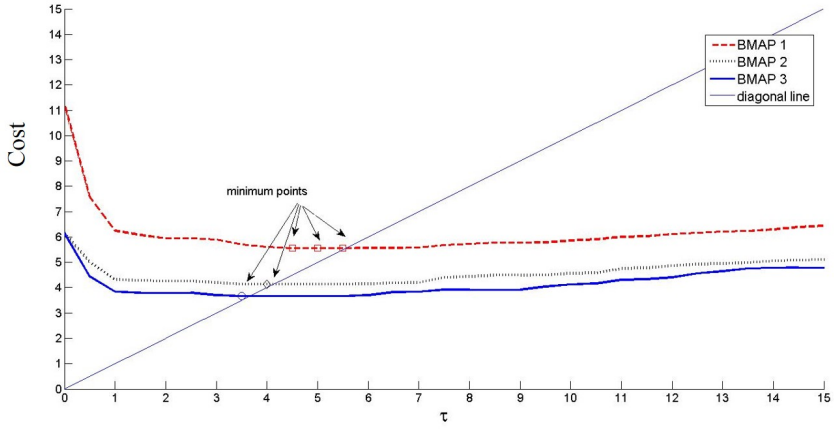

Figure 6.1: Costs of Threshold Policies in Example 6.2.1

Note: The input process in Example 2.1.1.(i) is a compound renewal input process, where there is only a single underlying state. In this case, the optimal delay-penalty threshold policy is actually the overall optimal policy. Next, we give a mathematical proof of its optimality.

Since there is only a single underlying state, we see that Ω and Φ are equivalent, and we can write the clearing function as $r(x)$. Consider two arbitrary stationary clearing policies r and r'. These two policies are uniquely identifiable by their corresponding post-clearing system state spaces $\Phi_{(r)}$ and $\Phi_{(y,r')}$. Define

$$\Phi^{+}_{r,r'} = \{y \in \Phi : \ r(y) = 1, \ r'(y) = 0\} \ \text{ and } \Phi^{-}_{r,r'} = \{y \in \Phi : \ r(y) = 0, r'(y) = 1\}. \quad (6.40)$$

It is easy to see that $\Phi^+_{r,r'} \in \Phi \setminus \Phi_{(r)}$, $\Phi^-_{r,r'} \in \Phi_{(r)}$, and

$$\Phi_{(y,r')} = \Phi_{(r)} \cup \Phi^+_{r,r'} \setminus \Phi^-_{r,r'}. \tag{6.41}$$

We shall call $\Phi^+_{r,r'}$ and $\Phi^-_{r,r'}$ "state space modifications" from r to r'.

Now we are ready to determine the overall optimal policy under a compound renewal input process and Condition 5.2.1.

Proposition 6.2.4. *For any two clearing policies r and r', we have*

$$\mathcal{C}(r) - \mathcal{C}(r') = \frac{\sum\limits_{x \in \Phi^+_{r,r'}} (\mathcal{C}(r) - H(x))\, R(x) + \sum\limits_{z \in \Phi^-_{r,r'}} (H(z) - \mathcal{C}(r))\, R(z)}{\sum\limits_{y \in \Phi_{(r)}} R(y) + \sum\limits_{x \in \Phi^+_{r,r'}} R(x) - \sum\limits_{z \in \Phi^-_{r,r'}} R(z)}, \tag{6.42}$$

where $\Phi^+_{r,r'}$ and $\Phi^-_{r,r'}$ are defined in Equation (6.40).

Proof. Since $\mathcal{C}(r) = \mathcal{C}_{\mathrm{K}}(r) + \mathcal{C}_{\mathrm{H}}(r)$, we will look at each component separately. First we note that

$$\sum_{y \in \Phi_{(y,r')}} R(y) = \sum_{y \in \Phi_{(r)}} R(y) + \sum_{x \in \Phi^+_{r,r'}} R(x) - \sum_{z \in \Phi^-_{r,r'}} R(z)$$

According to Equations (6.15), (6.16), (6.20), and (6.34), we have

$$\begin{aligned}
\mathcal{C}_{\mathrm{K}}(r) - \mathcal{C}_{\mathrm{K}}(r') &= \hat{k}[\theta(0)(1-d_0) - \theta'(0)(1-d_0)] \\
&= \hat{k}(1-d_0) \left(\frac{1}{\sum\limits_{y \in \Phi_{(r)}} R(y)} - \frac{1}{\sum\limits_{y \in \Phi_{(r)}} R(y) + \sum\limits_{x \in \Phi^+_{r,r'}} R(x) - \sum\limits_{z \in \Phi^-_{r,r'}} R(z)} \right) \\
&= \frac{\mathcal{C}_{\mathrm{K}}(r) \left(\sum\limits_{x \in \Phi^+_{r,r'}} R(x) - \sum\limits_{z \in \Phi^-_{r,r'}} R(z) \right)}{\sum\limits_{y \in \Phi_{(r)}} R(y) + \sum\limits_{x \in \Phi^+_{r,r'}} R(x) - \sum\limits_{z \in \Phi^-_{r,r'}} R(z)};
\end{aligned}$$

and from Equations (6.12), (6.13) and (6.36), we have

$$
\begin{aligned}
&\mathcal{C}_{\mathrm{H}}(r) - \mathcal{C}_{\mathrm{H}}(r') \\
&= \sum_{y\in\Phi_{(r)}} \mathcal{D}_{\mathrm{p}}(y)\theta(y) - \sum_{y\in\Phi_{(y,r')}} \mathcal{D}_{\mathrm{p}}(y)\theta'(y) \\
&= \frac{\sum\limits_{y\in\Phi_{(r)}} \mathcal{D}_{\mathrm{p}}(y)R(y)}{\sum\limits_{y\in\Phi_{(r)}} R(y)} - \frac{\sum\limits_{y\in\Phi_{(r)}} \mathcal{D}_{\mathrm{p}}(y)R(y) + \sum\limits_{x\in\Phi^{+}_{r,r'}} H(x)R(x) - \sum\limits_{z\in\Phi^{-}_{r,r'}} H(z)R(z)}{\sum\limits_{y\in\Phi_{(r)}} R(y) + \sum\limits_{x\in\Phi^{+}_{r,r'}} R(x) - \sum\limits_{z\in\Phi^{-}_{r,r'}} R(z)} \\
&= \frac{\mathcal{C}_{\mathrm{H}}(r)\left(\sum\limits_{x\in\Phi^{+}_{r,r'}} R(x) - \sum\limits_{z\in\Phi^{-}_{r,r'}} R(z)\right) + \left(\sum\limits_{z\in\Phi^{-}_{r,r'}} H(z)R(z) - \sum\limits_{x\in\Phi^{+}_{r,r'}} H(x)R(x)\right)}{\sum\limits_{y\in\Phi_{(r)}} R(y) + \sum\limits_{x\in\Phi^{+}_{r,r'}} R(x) - \sum\limits_{z\in\Phi^{-}_{r,r'}} R(z)}.
\end{aligned}
$$

Summing the two components and rearranging the terms will lead to Equation (6.42). □

Proposition 6.2.5. *Under a compound renewal input process and Condition 5.2.1, $\mathcal{C}(\tau)$ is unimodal for $0 \le \tau \le \hat{k}$ and $\tau^* = \mathcal{C}(\tau^*)$.*

Proof. If $\tau < \mathcal{C}(\tau)$ for some delay-penalty threshold policy parameterized by τ, let us define another policy with $\tau' = \tau + \epsilon$, where $0 < \epsilon < \mathcal{C}(\tau) - \tau$. For notational convenience, we shall use $\Phi_{(y,\tau)}$ and $\Phi_{(y,\tau')}$ to represent their corresponding post-clearing system state spaces. Note that $\Phi_{(y,\tau)} \subset \Phi_{(y,\tau')}$, so $\Phi^{-}_{\tau,\tau'} = \emptyset$, and $H(x) \le \mathcal{C}(\tau)$ for all $x \in \Phi^{+}_{\tau,\tau'} \subset \Phi_{(y,\tau)}$. By Proposition 6.2.4, we have

$$
\mathcal{C}(\tau) - C(r^{\tau'}) = \frac{\sum\limits_{x\in\Phi^{+}_{\tau,\tau'}} \left[\mathcal{C}(\tau) - H(x)\right] R(x)}{\sum\limits_{y\in\Phi_{(y,\tau)}} R(y) + \sum\limits_{x\in\Phi^{+}_{\tau,\tau'}} R(x)} \ge 0.
$$

Now let $\tau'' = \tau - \epsilon$. Then $\Phi_{(\tau'')} \subset \Phi_{(y,\tau)}$, $\Phi^{+}_{\tau,\tau''} = \emptyset$, and $H(z) \le \mathcal{C}(\tau)$ for all $z \in \Phi^{-}_{\tau,\tau''}$.

Again, by Proposition 6.2.4, we have

$$\mathcal{C}(\tau) - C(r^{\tau''}) = \frac{\sum\limits_{z\in\Phi^-_{\tau,\tau''}} (H(z) - \mathcal{C}(\tau))\, R(z)}{\sum\limits_{y\in\Phi_{(y,\tau)}} R(y) - \sum\limits_{z\in\Phi^-_{\tau,\tau''}} R(z)} \leq 0.$$

Thus, we have shown that $\mathcal{C}(\tau)$ is non-increasing when $\tau < \mathcal{C}(\tau)$. Similar arguments can be applied to show that $\mathcal{C}(\tau)$ is non-decreasing when $\tau > \mathcal{C}(\tau)$. Hence the function $\mathcal{C}(\tau)$ is minimized at τ^* satisfying $\tau^* = \mathcal{C}(\tau^*)$. This completes our proof. □

In general, $\mathcal{C}(\tau)$ may be minimized in an interval containing τ^*, i.e., $\exists[\tau_1, \tau_2]$ such that $\tau^* \in [\tau_1, \tau_2]$ and $\mathcal{C}(\tau) = \mathcal{C}(\tau^*) = \tau^*$, $\forall \tau \in [\tau_1, \tau_2]$. See Example 6.2.1 for evidence.

Theorem 6.2.6. *Under a compound renewal input process and Condition 5.2.1, the optimal delay-penalty threshold policy parameterized by τ^* is the overall optimal policy in terms of expected average total cost per period.*

Proof. For the optimal delay-penalty threshold policy $\tau^* = \mathcal{C}(\tau^*)$, we show that any state space modification to $\Phi_{(\tau^*)}$ will result in higher expected cost. Let r be such a policy; according to Equation (6.40), we can find the two sets $\Phi^+_{\tau^*,r}$ and $\Phi^-_{\tau^*,r}$. Note that $H(x) \geq \mathcal{C}(\tau^*)$ for all $x \in \Phi^+_{\tau^*,r}$ and $H(z) \leq \mathcal{C}(\tau^*)$ for all $z \in \Phi^-_{\tau^*,r}$. Then by Proposition 6.2.4, we have

$$\mathcal{C}(\tau^*) - \mathcal{C}(r) = \frac{\sum\limits_{x\in\Phi^+_{\tau^*,r}} (\mathcal{C}(\tau^*) - \mathcal{D}_{\mathrm{p}}(x))\, R(x) + \sum\limits_{z\in\Phi^-_{\tau^*,r}} (H(z) - \mathcal{C}(\tau^*))\, R(z)}{\sum\limits_{y\in\Phi_{(\tau^*)}} R(y) + \sum\limits_{x\in\Phi^+_{\tau^*,r}} R(x) - \sum\limits_{z\in\Omega^-_{\tau^*,r}} R(z)} \leq 0$$

This completes the proof. □

Based on the unimodularity observation on $\mathcal{C}(\tau)$, the following heuristic algorithm to search for τ^* is introduced. Note that in many cases, it is safe to assume that $0 \leq \tau^* \leq \hat{k}$,

because intuition suggests that delay penalty should not be allowed to exceed the fixed clearing cost. However, in the following algorithm, we can always extend our search range beyond $\hat{k}$ to be accurate.

Heuristic A: Delay Penalty Threshold Heuristic

A.1 Initialize $\varphi = 2 - \frac{1+\sqrt{5}}{2}$, $a = 0$, $b = \varphi\hat{k}$, $c = \hat{k}$ and choose a precision factor ϵ;

A.2 If $c - a < \epsilon$, STOP and RETURN $\tau^* = (c - a)/2$;

A.3 If $c - b > c - a$, set $\eta = b + \varphi(c - b)$; else set $\eta = b - \varphi(b - a)$;

A.4 Compute $\mathcal{C}(\eta)$ and $\mathcal{C}(b)$ according to Algorithm IX;

A.5 If $\mathcal{C}(\eta) < \mathcal{C}(b)$ and $c - b > b - a$, set $a = b$, $b = \eta$, $c = c$; else if $\mathcal{C}(\eta) < \mathcal{C}(b)$ and $c - b \leq b - a$, set $a = a$, $b = \eta$, $c = b$; else if $\mathcal{C}(\eta) \geq \mathcal{C}(b)$ and $c - b > b - a$, set $a = a$, $b = b$, $c = \eta$; else, set $a = \eta$, $b = b$, $c = c$. Go back to Step A.2;

Note that Algorithm II is a golden ratio search algorithm over the range $[0,\ \hat{k}]$. The total number of iterations is $\mathcal{O}(\log \frac{\hat{k}}{\epsilon})$. In each iteration, the bulk of the work lies in Step A.4, which has time complexity of $\mathcal{O}(Q^{\bar{L}})$. Therefore, the overall time complexity of this algorithm is $\mathcal{O}(Q^{\bar{L}} \log \frac{\hat{k}}{\epsilon})$.

6.3 Summary of Results

Modelling the stochastic clearing process as a GI/M/1 type Markov chain with a tree structure allows us to compute the expected average cost per period over an infinite horizon more efficiently. The stationary distribution of the system state can also be used to construct formulas to compute other performance measures such as the average delay per unit

input, average quantity cleared per clearing cycle, and average number of inputs cleared per clearing cycle.

By introducing some additional condition, we show that the optimal clearing policy can again be a threshold policy, and we introduce a heuristic algorithm to compute the optimal threshold level for expected average cost per period.

Chapter 7

Concluding Remarks

To conclude this thesis, we give a brief summary of the main results we have obtained through our research, followed by a discussion of some future research directions on stochastic clearing systems and other related systems.

7.1 Thesis Summary

In this thesis, we have studied stochastic clearing systems, which generalize a variety of stochastic systems found in different applications. These include shipment consolidation, inventory backlog, shuttle dispatch, and bulk service queues. We have conducted an extensive literature review to identify the similar modelling approaches and solution techniques in these areas. Through several accounts of practical applications, we have demonstrated the importance of studying stochastic clearing systems.

From the modelling perspective, one of the main contributions of our research is modelling the delay penalty cost as a function of both the quantity and age of individual inputs. In the previous literature, such a function is modelled as a function only of cumulative input quantity. That function has generally been linear, hence constant in time. Our approach does add more complexity to the models, but it allows the decision maker to take the

input age (i.e., the time a unit of input has spent in the system before being cleared) into consideration when deciding whether to continue accumulating inputs or clear the system. As we have shown through our examples, the input age can have significant impact on the performance and effectiveness of a clearing system, in terms of the service (i.e., the lead time) to the customer that will receive the respective input.

The other special feature of our models is the Markovian input/arrival process. Note that a Markov process has long been proven to be capable of approximating any input or arrival process [37]. By assuming that the input arrival rate varies according to an underlying state-of-the-world process, we can capture the potential correlations, burstiness, and other special characteristics of the actual input process. Markovian input processes are rarely considered in models of stochastic clearing systems, but they have already been used in many inventory and shipment consolidation models. We generalized these models in the context of stochastic clearing problems.

To accommodate the special model features, we record the system state as (x, i), where x is a sequence of numbers and i is the underlying state. The resulting system state space is very large, so to make our models mathematically tractable, we assume only discrete time and discrete quantity. Working with such a complicated system state space has its challenges, but it is a novel approach to represent the system state as a convolution of multiple dimensions of information.

From the solution perspective, this thesis contains complete solutions to three different stochastic clearing problems. These are the problems of expected total cost over a finite horizon, expected total discounted cost over an infinite horizon, and expected average total cost per period over an infinite horizon. For each of these problems, we have proven some analytical results and made some observations through numerical examples about the clearing policies and costs. These results and observations provide important insights on how to optimize the clearing policy.

The two main objectives of our studies are evaluating the performance of any given clearing policy, and finding the optimal clearing policy for a stochastic clearing system. Therefore, we use both Markov Decision Process (MDP) and Matrix-Analytic Methods (MAM) to construct multiple policy evaluation procedures and optimization algorithms and heuristics. We estimate and verify the correctness and complexity of these procedures, algorithms, and heuristics through numerical examples. MDP is effective in finding the optimal clearing policy, while MAM is efficient in policy evaluation and in optimization heuristics.

This thesis is completed with tables and figures of numerical examples and the MATLAB codes for all the procedures, algorithms, and heuristics. Through these extensive numerical analyses, we have made several observations and gained some insights on how to effectively manage a stochastic clearing system under special circumstances.

7.2 Future Research

The results presented in this thesis are only the beginning of what we see as a series of studies on stochastic clearing systems. There are several potential variations and improvements to our current models; some important theoretical questions remain unanswered. Below are a few of the research topics we intend to explore in the future.

7.2.1 Optimality of State-Dependent Threshold Clearing Policy

We have shown that, in the cases of expected total cost in a finite horizon and expected total discounted cost over an infinite horizon, if the cost functions satisfy certain conditions, then the optimal clearing policy can be a state-dependent threshold policy. However, in the case of expected average total cost per period over an infinite horizon, we are only able to speculate, through numerical examples, that the optimal policy can be of that form.

An analytical proof still eludes us. We hope to complete that proof in the future, and to identify the necessary and sufficient conditions for the optimal policy to be a state-dependent threshold policy.

7.2.2 Other Forms of Cost Structures

In this thesis, we have considered three types of costs associated with the stochastic clearing process. These are the delay penalty cost, the fixed clearing cost, and the variable clearing cost. In our analysis, we have made some restrictive assumptions or conditions on these cost functions. For instance, most of our results require that the variable clearing cost is a linear function of the total quantity cleared. However, in practice, the variable clearing cost can be subjected to an incremental or all-unit quantity discount. We plan to study the impact of such cost functions on the optimal clearing policy. A potential question here is whether the quantity discount creates an incentive for the decision maker to accumulate more inputs before clearing. That may depend on the customer disutility of waiting, i.e., on the delay penalty function.

In terms of that function, so far we have used only simple hypothetical functions to account for both the input quantities and ages. However, there is a lack of empirical or economic evidence to support a specific function that is increasing in an input's age. We hope to collect data or use intertemporal utility theory to find such functions that best fit the way that delay penalties affect a customer in practice.

7.2.3 Impact of Input Process

Intuition tells us that the clearing process will be most effective if inputs arrive frequently in relatively small sizes. Since we are modeling the input process with a MAP, which is known to be able to approximate any arrival process to a high precision, we will test

this conjecture with our models. We can do so by comparing systems whose order-arrival processes can be ranked under certain stochastic orders (see [73] and [57]).

Using the analytical results and the algorithms we have developed, we will also try to prove the preceding conjecture formally. As well, we are interested in determining the impact that stochastic orders of the input processes have on the parameters of the optimal clearing policy. For instance, if the optimal policy is proven to be a state-dependent threshold policy, how will the stochastic orders of the input processes affect the threshold levels? Lastly, regarding state-dependent threshold policies, it remains to be seen if the stochastic orders of the input quantities for different underlying states have any direct correlation with the orders of the clearing thresholds for the respective underlying states.

7.2.4 Partially-Observable or Unobservable Underlying States

We have successfully argued that the input process may depend on the underlying state of the world, and that the optimal clearing policy depends on this underlying state. However, what happens if the underlying state cannot be observed directly or at all by the decision maker? In that case, we can no longer rely on the state-dependent optimal clearing policy. However, we do know that the underlying states still play a role in the clearing process. In the field of Markov Decision Processes, such a process can be described as a "partially observable MDP"; the underlying states are estimated through past states and a belief-update function. If we can model that belief-update function, we can model the stochastic clearing process as a partially observable MDP.

Another potential way to deal with thess problems is to use the stationary distribution of the underlying process. In the long run, that stationary distribution of the underlying process gives the probability to be in a particular state. If we perform a random sampling of the stationary distribution at each decision epoch to "guess" the underlying state, we may still be able to use the optimal state-dependent clearing policy to achieve relatively

good performance.

7.2.5 Continuous Time and/or Continuous Quantity Models

Lastly, we will try to build new stochastic clearing models in continuous time and/or continuous quantity which are analogous to our discrete models. Although it may be more difficult and require advanced tools in probability theory, matrix analytical methods, stochastic calculus, etc., we can start off by improving upon the existing continuous models in inventory theory and shipment consolidation.

APPENDICES

Appendix A

Linear Programming Formulation for Total Discounted Cost Model

The problem of expected total discounted cost over an infinite horizon can also be formulated and solved as a pair of primal and dual linear programs. First let us define a clearing rule

$$r^{-}(x,i) = \begin{cases} 0, & \text{if } (x,i) \in \Omega_{\mathrm{U}} \\ 1, & \text{otherwise} \end{cases} \tag{A.1}$$

and another clearing rule

$$r^{+}(x,i) = 1, \forall x \in \Phi \setminus \{\emptyset\}. \tag{A.2}$$

The first rule is essentially following Proposition 4.3.4 to delay clearing as long as possible, while the second rule immediately clears any outstanding inputs.

Since we have a finite discrete system state space Ω_{U}, if no partial clearing is allowed, we can use these two rules to formulate the following linear program and its dual problem.

Primal Linear Program

$$\text{Maximize} \sum_{(x,i)\in\Omega_{\mathrm{U}}} \beta(x,i)V(x,i) \tag{A.3}$$

subject to

$$\begin{aligned} V(x,i) - \sum_{(y,j)\in\Omega_{\mathrm{U}}} \alpha P^{(x,r^-)}_{(x,i),(y,j)} V(y,j) &\leq u^{r^-}(x,i), \forall (x,i) \in \Omega_{\mathrm{U}} \\ V(x,i) - \sum_{(y,j)\in\Omega_{\mathrm{U}}} \alpha P^{(x,r^+)}_{(x,i),(y,j)} V(y,j) &\leq u^{r^+}(x,i), \forall (x,i) \in \Omega_{\mathrm{U}} \end{aligned} \tag{A.4}$$

and primal variables $V(x,i)$ are unconstrained.

Note that we can choose any set of positive scalars $\{\beta(x,i), \forall (x,i) \in \Omega_{\mathrm{U}}\}$, such that $\sum_{(x,i)\in\Omega_{\mathrm{U}}} \beta(x,i) = 1$. In fact, the set corresponds to the initial distribution of the system state. For the sake of simplicity, we can set $\beta(x,i) = \frac{1}{|\Omega_{\mathrm{U}}|}$ uniformly. The transition probabilities $P^{(x,r^-)}_{(x,i),(y,j)}$ and $P^{(x,r^+)}_{(x,i),(y,j)}$ are given by Equations (2.25) to (2.27), and the functions $u^{r^-}(x,i)$ and $u^{r^+}(x,i)$ are defined by Equation (3.2).

Dual Linear Program

$$\text{Minimize} \sum_{(x,i)\in\Omega_{\mathrm{U}}} u^{r^-}(x,i) Z^-(x,i) + u^{r^+}(x,i) Z^+(x,i) \tag{A.5}$$

subject to

$$\sum_{(y,j)\in\Omega_{\mathrm{U}}} Z^-(x,i) \left[1 - \alpha P^{(x,r^-)}_{(x,i),(y,j)}\right] + Z^+(x,i) \left[1 - \alpha P^{(x,r^+)}_{(x,i),(y,j)}\right] = \beta(x,i), \forall (x,i) \in \Omega_{\mathrm{U}} \tag{A.6}$$

and dual variables $Z^-(x,i) \geq 0$ and $Z^+(x,i) \geq 0$.

The dual variables $Z^-(x,i)$ and $Z^+(x,i)$, for all $(x,i) \in \Omega_U$, represents the two possible actions to choose from when the system is in state (x,i). Thye directly correspond to the clearing rule r. The dual problem contains more useful information. We usually choose to solve it instead of the primal because the former involves fewer constraints. Suppose we collect all states in Ω_{U} and denote the solution to the dual LP by a vector Z which contains both Z^- and Z^+. It is proven in [68] that, if Z is a basic feasible solution to the

dual LP, then we can create an MD policy by

$$r^*(x,i) = \begin{cases} 0, & \text{if } \frac{Z^-(x,i)}{Z^-(x,i)+Z^+(x,i)} = 1 \\ 1, & \text{otherwise} \end{cases} \tag{A.7}$$

The following theorem establishes the relationship between optimal policies and optimal solutions of the dual LP.

Theorem A.0.1. *(Theorem 6.9.4 in [68]) For the dual LP of the infinite horizon problem, the following statements are equivalent*

(i) There exists a bounded optimal basic feasible solution Z^* to the dual LP.

(ii) Z^* is an optimal solution to the dual LP, if and only if the corresponding policy r_* is an optimal policy.

(iii) Z^* is an optimal basic solution to the dual LP, if and only if the corresponding policy r_* is an optimal MD policy.

Appendix B

Linear Programming Formulation for Average Total Cost Model

Recall from Section 4.4 that we can create two clearing policies as

$$r^{-}(x,i) = \begin{cases} 0, & \text{if } H(x) \leq \hat{k} \\ 1, & \text{otherwise} \end{cases}$$

and

$$r^{+}(x,i) = 1, \forall x \in \Phi \setminus \{\emptyset\}.$$

The first rule is essentially following Lemma 5.2.1 to delay clearing as long as possible, while the second rule immediately clears any outstanding inputs.

Since we have a finite discrete system state space Ω_{U}, if no partial clearing is allowed, we can use these two rules to formulate the following linear program and its dual problem.

Primal Linear Program

$$\text{Maximize} \sum_{(x,i)\in\Omega_{\mathrm{U}}} \beta(x,i)V(x,i) \tag{B.1}$$

subject to

$$\begin{aligned} g + h(x,i) - \sum_{(y,j)\in\Omega_{\mathrm{U}}} \alpha P^{(x,r^-)}_{(x,i),(y,j)} h(y,j) &\leq u^{r^-}(x,i), \forall (x,i) \in \Omega_{\mathrm{U}} \\ g + h(x,i) - \sum_{(y,j)\in\Omega_{\mathrm{U}}} \alpha P^{(x,r^+)}_{(x,i),(y,j)} h(y,j) &\leq u^{r^+}(x,i), \forall (x,i) \in \Omega_{\mathrm{U}} \end{aligned} \tag{B.2}$$

with primal variables g and $h(x,i)$ unconstrained.

Note that we can choose any set of positive scalars $\{\beta(x,i), \forall (x,i) \in \Omega_{\mathrm{U}}\}$, such that $\sum_{(x,i)\in\Omega_{\mathrm{U}}} \beta(x,i) = 1$. In fact, the set corresponds to the initial distribution of the system state. For the sake of simplicity, we can set $\beta(x,i) = \frac{1}{|\Omega_{\mathrm{U}}|}$ uniformly. The transition probabilities $P^{(x,r^-)}_{(x,i),(y,j)}$ and $P^{(x,r^+)}_{(x,i),(y,j)}$ are given by Equations (2.25) to (2.27), and the functions $u^{r^-}(x,i)$ and $u^{r^+}(x,i)$ are defined by Equation (3.2).

Dual Linear Program

$$\text{Minimize} \sum_{(x,i)\in\Omega_{\mathrm{U}}} u^{r^-}(x,i) Z^-(x,i) + u^{r^+}(x,i) Z^+(x,i) \tag{B.3}$$

subject to

$$\begin{aligned} &\sum_{(y,j)\in\Omega_{\mathrm{U}}} Z^-(x,i) \left[1 - \alpha P^{(x,r^-)}_{(x,i),(y,j)}\right] + Z^+(x,i) \left[1 - \alpha P^{(x,r^+)}_{(x,i),(y,j)}\right] = \beta(x,i), \forall (x,i) \in \Omega_{\mathrm{U}} \\ &\sum_{(y,j)\in\Omega_{\mathrm{U}}} Z^-(x,i) + Z^+(x,i) = 1 \end{aligned} \tag{B.4}$$

and dual variables $Z^-(x,i) \geq 0$ and $Z^+(x,i) \geq 0$.

Suppose we collect all states in Ω_{U} and denote the solution to the dual LP by a vector Z which contains both Z^- and Z^+. It is proven in [68] that, if Z^* is a bounded basic

feasible solution to the dual LP, then we can create an optimal MD policy by

$$r^*(x,i) = \begin{cases} 0, & \text{if } Z^-(x,i) > 0 \\ 1, & \text{if } Z^+(x,i) > 0 \\ \text{arbitrary}, & \text{otherwise} \end{cases} \tag{B.5}$$

Appendix C

Complexity Studies

In this appendix section, we present numerical examples to verify the time and memory complexity estimates for all the procedures, algorithms, and heuristics we have constructed in this thesis. Some procedures and algorithms have similar complexity and use the same parameters, so we test them with the same numerical examples. We summarize the complexity estimates in Table C.1.

Order of Growth	Time Complexity	Memory Complexity	Convergence
BFT Procedures I and II	$\mathcal{O}(MQ^{\bar{L}_{(r_t)}})$	$\mathcal{O}(\bar{L}_{(r_t)}Q^{\bar{L}_{(r_t)}})$	NA
Algorithm I	$\mathcal{O}(M^2NQ^{\bar{L}_{(r_t)}})$	$\mathcal{O}(\bar{L}_{(r_t)}Q^{\bar{L}_{(r_t)}})$	NA
Algorithm II	$\mathcal{O}(M^2NQ^{\tilde{L}})$	$\mathcal{O}(\tilde{L}NQ^{\tilde{L}})$	NA
Algorithm III	$\mathcal{O}(M^3Q^{\bar{L}_{(r)}})$	$\mathcal{O}(M^2Q^{\bar{L}_{(r)}})$	NA
Algorithm IV	$\mathcal{O}(M^3Q^{\tilde{L}})$	$\mathcal{O}(M\tilde{L}Q^{\tilde{L}})$	linear
Algorithm V	$\mathcal{O}(M^3Q^{\tilde{L}})$	$\mathcal{O}(M\tilde{L}Q^{\tilde{L}})$	linear
Algorithm VI	$\mathcal{O}(M^3Q^{\bar{L}_{(r)}})$	$\mathcal{O}(M^2Q^{\bar{L}_{(r)}})$	NA
Algorithm VII	$\mathcal{O}(M^3Q^{\tilde{L}})$	$\mathcal{O}(M\tilde{L}Q^{\tilde{L}})$	finite
Algorithm VIII	$\mathcal{O}(M^3Q^{\tilde{L}})$	$\mathcal{O}(M\tilde{L}Q^{\tilde{L}})$	finite
Stationary Distn Procedure	$\mathcal{O}(MQ^{\bar{L}_{(r)}})$	$\mathcal{O}(M^2Q^{\bar{L}_{(r)}})$	NA
Algorithm IX	$\mathcal{O}(Q^{\bar{L}_{(r)}})$	$\mathcal{O}(M^2Q^{\bar{L}_{(r)}})$	NA

Table C.1: Complexity Summary

Note that the complexity estimates may depend on the length of the planning horizon N, the number of underlying states M, the maximum input quantity Q, and $\bar{L}_{(r)}$, the

maximum input age allowed, or $\tilde{L}$, the upper bound of maximum input age allowed. The major components in all of the complexity estimates are $Q^{\bar{L}_{(r)}}$ or $Q^{\tilde{L}}$, which suggest exponential order of growth. In comparison, N and M are usually relatively small in practice. Therefore, in our numerical examples, we focus on testing different values of Q and $\bar{L}_{(r)}$ or $Q^{\tilde{L}}$.

Note that BFT Procedure I is used in all of Algorithms, and that procedure accounts for the bulk of the work in all of these algorithms. Without loss of generality, let us assume that the clearing policy is stationary and is given as

$$r(x_t, i_t) = \begin{cases} 0, & \text{if } \mathcal{L}(x_t) < \bar{L}_{(r)} \\ 1, & \text{otherwise} \end{cases} .$$

The other parameters are given as follows. For the finite horizon problem, let $N = 10$. The input process is given as

$$D_0 = \begin{pmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{pmatrix}, D_1 = \frac{1}{Q}\begin{pmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{pmatrix}, \ldots, D_Q = \frac{1}{Q}\begin{pmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{pmatrix}.$$

We record the run time (in seconds) and the size of $\Psi_{(r)}$ or $\Phi_{(r)}$ for BFT Procedure I in the following tables and plots. The numerical results provide sufficient evidence to support our estimate of the time and memory complexities for the procedure. The complexities for the policy evaluation algorithms scale proportionally to BFT Procedure I.

Run time(sec)	$\bar{L}_{(r)} = 1$	$\bar{L}_{(r)} = 2$	$\bar{L}_{(r)} = 3$	$\bar{L}_{(r)} = 4$	$\bar{L}_{(r)} = 5$
$Q = 1$	$3.8772e^{-4}$	$7.6347e^{-4}$	$9.3852e^{-4}$	0.0013	0.0024
$Q = 2$	$2.2066e^{-4}$	$5.0803e^{-4}$	0.0016	0.0049	0.0158
$Q = 3$	$1.8588e^{-4}$	$8.5299e^{-4}$	0.0039	0.0165	0.0888
$Q = 4$	$2.1040e^{-4}$	0.0013	0.0075	0.0483	0.4518
$Q = 5$	$2.6228e^{-4}$	0.0019	0.0137	0.1269	2.3030

Table C.2: BFT Procedure I Time Complexity Tests

$\lvert\Psi_{(r)}\rvert$	$\bar{L}_{(r)}=1$	$\bar{L}_{(r)}=2$	$\bar{L}_{(r)}=3$	$\bar{L}_{(r)}=4$	$\bar{L}_{(r)}=5$
$Q=1$	2	4	8	16	32
$Q=2$	3	9	27	81	243
$Q=3$	4	16	64	256	1024
$Q=4$	5	25	125	625	3125
$Q=5$	6	36	216	1296	7776

Table C.3: BFT Procedure I Memory Complexity Tests

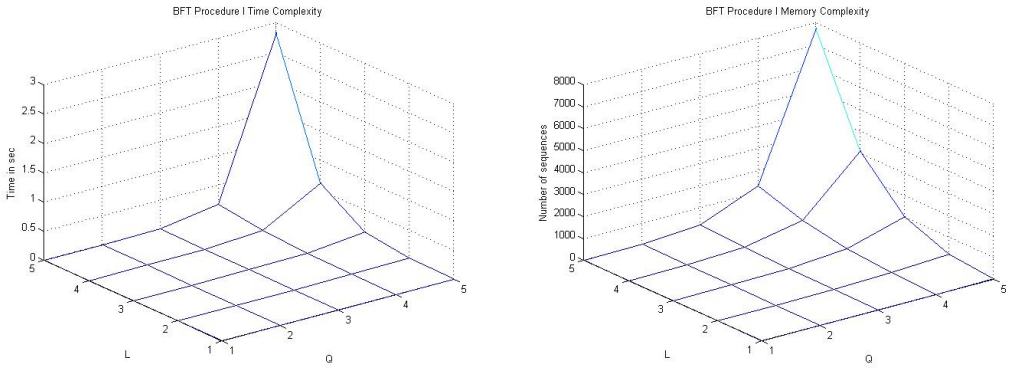

Figure C.1: BFT Procedure I Complexity Tests

References

[1] J.R. Artalejo and A. Gomez-Corral. Analysis of a stochastic clearing system with repeated attempts. *Stochastic Models*, 14(3):623–645, 1998.

[2] F. Barbera, H. Schneider, and P. Kelle. A condition based maintenance model with exponential failures and fixed inspection intervals. *Journal of the Operational Research Society*, 47(8):1037–1045, 1996.

[3] R. Bellman, I. Glicksberg, and O. Gross. On the optimal inventory equation. *Management Science*, 2(1):83–104, 1955.

[4] R.E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.

[5] D. Bertsekas. *Dynamic Programming and Stochastic Control*. Academic Press, New York, New York, 1976.

[6] D. Bertsekas and S.E. Shreve. *Stochastic Optimal Control: the Discrete Time Case*. Academic Press, New York, New York, 1976.

[7] D. Beyer, F. Cheng, S.P. Sethi, and M. Taksar. *Markov Demand Inventory Models*. Springer US, New York, New York, 2010.

[8] D. Beyer and S.P. Sethi. Average cost optimality in inventory models with Markovian demands. *Journal of Optimization Theory and Applications*, 92(3):497–526, 1997.

[9] D. Beyer and S.P. Sethi. The classical average-cost inventory models of Iglehart and Veinott-Wagner revisited. *Journal of Optimization Theory and Applications*, 101(3):523–555, 1999.

[10] D. Beyer, S.P. Sethi, and R. Sridhar. Stochastic multiproduct inventory models with limited storage. *Journal of Optimization Theory and Applications*, 111(3):553–588, 2001.

[11] D. Beyer, S.P. Sethi, and M. Taksar. Inventory models with Markovian demands and cost functions of polynomial growth. *Journal of Optimization Theory and Applications*, 98(2):281–323, 1998.

[12] J.H. Bookbinder, Q. Cai, and Q.M. He. Shipment consolidation by private carrier: the discrete time and discrete quantity case. *Stochastic Models*, 27(4):664–686, 2011.

[13] J.H. Bookbinder and J.K. Higginson. Probabilistic modeling of freight consolidation by private carriage. *Transportation Research, Part E*, 38(5):305–318, 2002.

[14] O.J. Boxma, D. Perry, and W. Stadje. Clearing models for M/G/1 Queues. *Queueing Systems*, 38(3):287–306, 2001.

[15] Q. Cai, Q.M. He, and J.H. Bookbinder. A tree-structured Markovian model of the shipment consolidation process. *Stochastic Models*, 30(4):521–553, 2014.

[16] S. Çetinkaya and J.H. Bookbinder. Stochastic models for the dispatch of consolidated shipments. *Transportation Research, Part B*, 37(8):747–768, 2003.

[17] S. Çetinkaya, E. Tekin, and C.Y. Lee. A stochastic model for joint inventory and outbound shipment decisions. *IIE Transactions*, 40(3):324–340, 2008.

[18] Sila Çetinkaya. *Applications of Supply Chain Management and E-Commerce Research*, chapter Coordination of inventory and shipment consolidation decisions: A review of premises, models, and justification. Springer US, New York, New York, 2005.

[19] F. Chen and J.S. Song. Optimal policies for multiechelon inventory problems with Markov-modulated demand. *Operations Research*, 49(2):226–234, 2001.

[20] F.Y. Chen, T. Wang, and T.Z. Xu. Integrated inventory replenishment and temporal shipment consolidation: A comparison of quantity-based and time-based models. *Annals of Operations Research*, 135(1):197–210, 2005.

[21] X. Chen and D. Simchi-Levi. Coordinating inventory control and pricing strategies with random demand and fixed ordering cost: The finite horizon case. *Operations Research*, 52(6):887–896, 2003.

[22] Y. Chen, S. Ray, and Y. Song. Optimal pricing and inventory control policy in periodic-review systems with fixed ordering cost and lost sales. *Naval Research Logistics*, 53(2):117–136, 2006.

[23] F. Cheng and S.P. Sethi. A periodic review inventory model with demand influenced by promotion decisions. *Management Science*, 45(11):1510–1523, 1999a.

[24] F. Cheng and S.P. Sethi. Optimality of state-dependent (s, S) policies in inventory models with Markov-modulated demand and lost sales. *Production and Operations Management*, 8(2):183–192, 1999b.

[25] Y.S. Chow, H. Robbins, and D. Siegmund. *Great Expectations: the Theory of Optimal Stopping*. Houghton Mifflin, Boston, Massachusetts, 1971.

[26] R.K. Deb. Optimal dispatching of a finite capacity shuttle. *Management Science*, 24(13):13621372, 1978.

[27] C. Derman. *Finite State Markovian Decision Processes*. Academic Press, Orlando, Florida, 1957.

[28] A. Drexl and A. Kimms. Lot sizing and scheduling - survey and extensions. *European Journal of Operational Research*, 99(2):221–235, 1997.

[29] M. Dror and B.C. Hartman. Shipment consolidation: Who pays for it and how much? *Management Science*, 53(1):78–87, 2007.

[30] A.N. Dudin and A.V. Karolik. BMAP/SM/1 queue with Markovian input of disasters and non-instantaneous recovery. *Performance Evaluation*, 45(1):19–32, 2001.

[31] A. Federgruen and P. Zipkin. An efficient algorithm for computing optimal (s, S) policies. *Operations Research*, 32(1):1268–1285, 1984.

[32] B. Fleischmann. The discrete lot-sizing and scheduling problem. *European Journal of Operational Research*, 44(3):337–348, 1990.

[33] Y.P. Gupta and P.K. Bagchi. Inbound freight consolidation under just-in-time procurement: Application of clearing models. *Journal of Business Logistics*, 8(2):74–94, 1987.

[34] J. Gutierrez, A. Sedeo-Noda, M. Colebrook, and J. Sicilia. A new characterization for the dynamic lot size problem with bounded inventory. *Computers and Operations Research*, 30(3):383395, 2003.

[35] Q.M. He. The classification of Markov chains of matrix GI/M/1 type with a tree structure and its queueing applications. *Journal of Applied Probability*, 40(4):1087–1102, 2003.

[36] Q.M. He. A fixed point approach to the classification of Markov chains with a tree structure. *Stochastic Models*, 19(1):76–114, 2003.

[37] Q.M. He. *Fundamentals of Matrix-Analytic Methods*. Springer, New York, New York, 2014.

[38] Q.M. He and M.F. Neuts. Markov chains with marked transitions. *Stochastic Processes and their Applications*, 74(1):37–52, 1998.

[39] M.D. Hickman. An analytic stochastic model for the transit vehicle holding problem. *Transportation Science*, 35(3):215237, 2001.

[40] J.K. Higginson. Recurrent decision approaches to shipment-release timing in freight consolidation. *International Journal of Physical Distribution and Logistics Management*, 25(5):3–23, 1995.

[41] J.K. Higginson and J.H. Bookbinder. Policy recommendations for a shipment consolidation program. *Journal of Business Logistics*, 15(1):87–112, 1994.

[42] J.K. Higginson and J.H. Bookbinder. Markovian decision processes in shipment consolidation. *Transportation Science*, 29(3):242–255, 1995.

[43] F.S. Hillier and G.J. Lieberman. *Introduction to Operations Research*. McGraw-Hill Education, New York, New York, 2015.

[44] W.T. Huh and G. Janakiraman. (s, S) optimality in joint inventory-pricing control: An alternative approach. *Operations Research*, 56(3):783–790, 2008.

[45] D.L. Iglehart. Optimality of (s, S) policies in the infinite horizon dynamic inventory problem. *Management Science*, 9(2):259–267, 1963.

[46] D.L. Iglehart and S. Karlin. *Optimal Policy for Dynamic Inventory Process with Nonstationary Stochastic Demands*, chapter Studies in Applied Probability and Management Science. Stanford University Press, Stanford, California, 1962.

[47] E. Ignall and P. Kolesar. Optimal dispatching of an infinite-capacity shuttle: Control at a single terminal. *Operations Research*, 22(5):10081024, 1974.

[48] S.D. Jacka. Optimal stopping and the American put. *Mathematical Finance*, 1(2):1–14, 1991.

[49] E.P.C. Kao. A multi-product dynamic lot-size model with individual and joint set-up costs. *Operations Research*, 27(2):279 – 289, 1979.

[50] I. Karatzas and S.E. Shreve. *Methods of Mathematical Finance*. Springer US, New York, New York, 1998.

[51] O. Kella, D. Perry, and W. Stadje. A stochastic clearing model with a Brownian and a compound Poisson component. *Probability in the Engineering and Informational Sciences*, 17(1):1–22, 2003.

[52] K. Kim and A.F. Seila. A generalized cost model for stochastic clearing systems. *Computers & Operations Research*, 20(1):67–82, 1993.

[53] W.K. Kruse. Technical note: Waiting time in a continuous review (s, S) inventory system with constant lead times. *Operations Research*, 29(1):202–207, 1981.

[54] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA and SIAM, Philadelphia, Pennsylvania, 1999.

[55] G. Latouche, M.A. Remiche, and P. Taylor. Transient Markov arrival processes. *The Annals of Applied Probability*, 13(2):628–640, 2003.

[56] D.M. Lucantoni. New results on the single server queue with a batch Markovian arrival process. *Stochastic Models*, 7(1):1–46, 1991.

[57] A. Marshall, I. Olkin, and B. Arnold. *Inequalities: Theory of Majorization and Its Applications*. Springer, New York, New York, 2010.

[58] F. Mutlu, S. Çetinkaya, and J.H. Bookbinder. An analytical model for computing the optimal time-and-quantity-based policy for consolidated shipments. *IIE Transactions*, 42(5):367–377, 2010.

[59] M.F. Neuts. A versatile Markovian point process. *Journal of Applied Probability*, 16(4):764–779, 1979.

[60] M.F. Neuts. *Matrix-geometric solutions in stochastic models - an algorithmic approach*. The John Hopkins University Press, Baltimore, Maryland, 1981.

[61] M.F. Neuts. *Structured Stochastic Matrices of M/G/1 type and their Applications*. Marcel Dekker, New York, New York, 1989.

[62] M.F. Neuts. Models based on the Markovian arrival process. *IEICE Trans. Commun*, 75(2):1255–1265, 1992.

[63] K.P. Papadaki and W.B. Powell. Exploiting structure in adaptive dynamic programming algorithms for a stochastic batch service problem. *European Journal of Operational Research*, 142(1):108127, 2002.

[64] K.P. Papadaki and W.B. Powell. An adaptive dynamic programming algorithm for a stochastic multiproduct batch dispatch problem. *Naval Research Logistics*, 50(7):742769, 2003.

[65] D.A. Popken. An algorithm for the multiattribute, multicommodity flow problem with freight consolidation and inventory costs. *Operations Research*, 42(2):274–286, 1994.

[66] E.L. Porteus. On the optimality of generalized (s, S) policies. *Management Science*, 17(7):411–426, 1971.

[67] E.L. Porteus. Optimal lot sizing, process quality improvement and setup cost reduction. *Operations Research*, 34(1):137–140, 1986.

[68] M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, Inc., Hoboken, New Jersey, 2005.

[69] S.M. Ross. *Introduction to Probability Models*. Elsevier, Kidlington, Oxford, 2010.

[70] H. Scarf. *Mathematical Methods in the Social Sciences*, chapter The optimality of (S, s) policies in the dynamic inventory problem. Stanford University Press, Stanford, California, 1960.

[71] R. Serfozo and S. Stidham. Semi-stationary clearing processes. *Stochastic Processes and their Applications*, 6(2):165–178, 1978.

[72] S.P. Sethi and F. Cheng. Optimality of (s, S) policies in inventory models with Markovian demand. *Operations Research*, 45(6):931–939, 1997.

[73] M. Shaked and J.G. Shanthikumar. *Stochastic Orders*. Springer, New York, New York, 2007.

[74] J. G. Shanthikumar and U. Sumita. General shock models associated with correlated renewal sequences. *Journal of Applied Probability*, 20(3):600–614, 1983.

[75] A.N. Shiryaev. *Optimal Stopping Rules*. Springer US, New York, New York, 2008.

[76] J.S. Song and P. Zipkin. Inventory control in a fluctuating demand environment. *Operations Research*, 41(2):351–370, 1993.

[77] S. Stidham. Stochastic clearing systems. *Stochastic Processes and their Applications*, 2(1):85–113, 1974.

[78] S. Stidham. Cost models for stochastic clearing systems. *Operations Research*, 25(1):100–127, 1977.

[79] S. Stidham. Clearing systems and (s, S) inventory systems with nonlinear costs and positive lead times. *Operations Research*, 34(2):276–280, 1986.

[80] J. Teghem. Control of the service process in a queueing system. *European Journal of Operational Research*, 23(2):141–158, 1986.

[81] J.C. Tyan, F.K. Wang, and T.C. Du. An evaluation of freight consolidation policies in global third party logistics. *Omega*, 31(1):55–62, 2003.

[82] A.F. Veinott. On the opimality of (s, S) inventory policies: New conditions and a new proof. *SIAM Journal on Applied Mathematics*, 14(5):1067–1083, 1966.

[83] A.F. Veinott and H.M. Wagner. Computing optimal (s, S) inventory policies. *Management Science*, 11(5):525–552, 1965.

[84] H.M. Wagner and T.M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5(1):89–96, 1958.

[85] H.J. Weiss. The computation of optimal control limits for a queue with batch services. *Management Sciences*, 25(4):320328, 1979.

[86] W. Whitt. The stationary distribution of a stochastic clearing process. *Operations Research*, 29(2):294–308, 1981.

[87] W.S. Yang, J.D. Kim, and K.C. Chae. Analysis of M/G/1 stochastic clearing systems. *Stochastic Analysis and Applications*, 20(5):1083–1100, 2002.

[88] R.W. Yeung and B. Sengupta. Matrix product-form solutions for Markov chains with a tree structure. *Advances in Applied Probability*, 26(4):965–987, 1994.

[89] W.I. Zangwill. A backlogging model and a multi-echelon model of a dynamic economic lot size production systema network approach. *Management Science*, 15(9):506 – 527, 1969.

[90] Y.S. Zheng. A simple proof for optimality of (s, S) policies in infinite-horizon inventory systems. *Journal of Applied Probability*, 28(4):802–810, 1991.

[91] Y.S. Zheng and A. Federgruen. Finding optimal (s, S) policies is about as simple as evaluating a single policy. *Operations Research*, 39(4):654–665, 1991.