INDEX POLICIES FOR PATIENT SCHEDULING AND ATM REPLENISHMENT

Yu Zhang

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill
2016

Approved by:

Vidyadhar Kulkarni

Nilay Argon

Jayashankar Swaminathan

Quoc Tran-Dinh

Serhan Ziya

ABSTRACT

YU ZHANG: INDEX POLICIES FOR PATIENT SCHEDULING AND ATM REPLENISHMENT
(Under the direction of Vidyadhar Kulkarni)

Markov Decision Processes (MDP) are one of the most commonly used stochastic models to solve sequential decision making problems. The optimal solution to many real-world problems cannot be achieved due to the curse of dimensionality. It is common to use a heuristic policy called the index policy, which is obtained by applying one-step policy improvement to a simple initial policy. The index policy performs close to the optimal policy and is easily implementable, which makes it attractive to use in practice. In this dissertation, we first introduce the background information on MDP and index policies in Chapter 1. We then study their applications in two problems: the appointment scheduling problem with patient preferences, and the automated teller machine (ATM) replenishment problem.

In Chapter 2, we build an MDP model to design appointment scheduling policies in the presence of patient preferences. We model the patient preferences by assuming that each patient has a set of appointment days that are equally acceptable to the patient. We consider a service provider which receives the appointment-booking requests and makes an appointment decision one at a time. The objective is to minimize the long-run average cost while responding to the patients' booking requests based on their preferences. We propose the index policy and show it performs close to the optimal policy in the two-day horizon and outperforms other benchmarks in the multi-day horizon.

In Chapter 3, we build an MDP model to design ATM replenishment schedules, while balancing the cost of replenishments and the cost of stock-outs. We propose a method to establish a relationship between the service level and the cost of a stock-out. We also assume that the replenishment cost is a sub-modular function of the set of ATMs that are replenished together. We derive the index policy, prove it has the same structural properties as the optimal policy, and show it performs close to the optimal policy when there are two or three ATMs. When there are a large number of ATMs, we show the index policy outperforms a benchmark policy through a simulation study and a real-world data-set.

ACKNOWLEDGMENTS

I want to express my deepest gratitude to my advisor, Professor Vidyadhar Kulkarni, for his teaching, guiding, supporting, and encouraging me during the past five years. I feel extremely fortunate to have him as my advisor. Learning from him and working with him made my doctoral study an enjoyable and rewarding experience. He not only taught me how to translate a real-world problem into a research project and build a mathematical model to solve it, but also how to communicate these ideas to others effectively. He helped me grow and improve myself continuously. My gratefulness for numerous efforts he made for me is beyond words.

I also want to extend my appreciation to other committee members: Professor Nilay Argon, Professor Jayashankar Swaminathan, Professor Quoc Tran-Dinh, and Professor Serhan Ziya. In particular, I am grateful to have taken Professor Nilay Argon's courses on Discrete Event Simulation and Markov Decision Processes and Professor Serhan Ziya's course on Design and Control of Queueing Systems. Without their effort explaining these complicated subjects in a clear and engaging way, I cannot leverage these useful methods in my research. I am also grateful to Professor Jayashankar Swaminathan and Professor Quoc Tran-Dinh for their insightful comments on my research.

I would like to thank Professor Onno Boxma for his valuable comments on the work related to appointment scheduling. I also would like to acknowledge SAS Institute Inc. for sparking my interest in the ATM replenishment problem and providing valuable feedback on the formulation of the problem.

Last but not least, I would like to thank my parents, Yuntao Zhang and Ying Li. My appreciation for their endless love and support is beyond description.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: Markov Decision Processes and Index Policies: Background

In this doctoral dissertation, we plan to use Markov Decision Processes and index policies to model and analyze two real-world problems. In this chapter, we collect the background information on Markov Decision Processes (MDP) and the index policies.

MDP is a tool to study sequential decision making problem. (Puterman, 2014) is an excellent reference on this subject. An MDP has five elements: decision epochs, a state space, an action space, transition probabilities and costs. A decision epoch is the point of time when a decision is made. Let X_n be the system state at the decision epoch n. Suppose $X_n \in S$ for all $n \geq 0$. We call S a state space. Let A_n be the action taken at the decision epoch n. Suppose $A_n \in \mathscr{A}$ for all $n \geq 0$. We call $\mathscr{A}$ an action space. The process $\{(X_n, A_n), n \geq 0\}$ is called an MDP if

$$\mathrm{P}(X_{n+1} = j | X_n = i, A_n = a, X_{n-1}, ..., X_1, X_0, A_{n-1}, ..., A_1, A_0) = p_{ij}(a),$$

for all $n \geq 0, i, j \in S, a \in \mathscr{A}$. We call $p_{ij}(a)$ transition probabilities. Let $c(i, a)$ be the expected cost incurred if action a is chosen in state i at any time $n \geq 0$.

A policy is a description of actions taken at each decision epoch. Let $\pi : S \to \mathscr{A}$ be a policy. We choose action $\pi(i)$ whenever the system is in state i under policy π. Such policies are called stationary deterministic Markovian (SDM) policies. For a given SDM policy π, define (assuming the limit exists)

$$g^{\pi}(i) = \lim_{N\to\infty} \frac{1}{N+1} \mathrm{E}^{\pi} \left[\sum_{n=0}^{N} c(X_n, A_n) \middle| X_0 = i \right].$$

We call $g^{\pi}(i)$ the long-run average cost of following the policy π. Let

$$g^*(i) = \inf_{\pi} g^{\pi}(i), \quad \forall i \in S.$$

If there is a policy π^* that achieves this infimum, it is called the average-cost optimal policy. Thus an optimal policy (if it exists) satisfies

$$g^{\pi^*}(i) = g^*(i), \quad \forall i \in S.$$

Now we discuss when such an optimal policy exists and how to compute it. Define

$$v_{n+1}(i) = \min_{a\in\mathscr{A}} \{c(i,a) + \sum_{j\in S} p_{ij}(a) v_n(j)\}, \tag{1.1}$$

for all $i \in S$ and $n \geq 0$, where $v_0(i) = 0$ all $i \in S$. We can interpret $v_n(i)$ as the optimal total expected cost incurred over the n days starting from state i. It is known (see (Tijms, 2003)) that $v_n(i)$ is asymptotically linear in n with slope g and intercept $h(i)$. We can write

$$v_n(i) = ng + h(i) + o(n), \tag{1.2}$$

where $\frac{o(n)}{n} \to 0$ as $n \to \infty$. The slope g is the optimal long-run average cost. The intercept

$h(\cdot)$ is called the bias function. It is known (see (Tijms, 2003)) that g and $h(\cdot)$ satisfy the Bellman equation

$$h(i) + g = \min_{a \in \mathscr{A}} \{c(i,a) + \sum_{j \in S} p_{ij}(a) h(j)\}, \tag{1.3}$$

It is also known (see (Tijms, 2003)) that if Equation (1.3) has a solution, then we can use it to compute the optimal decision as follows. Define

$$a(i) = \arg\min_{a \in \mathscr{A}} \{c(i,a) + \sum_{j \in S} p_{ij}(a) h(j)\}, \quad \forall i \in S. \tag{1.4}$$

The standard theory of dynamic programming shows that the Markovian policy that chooses action $a(i)$ in state i is optimal. It is known (see (Tijms, 2003)) that Equation (1.3) has a solution if the MDP is unichain, that is, for each stationary policy the associated Markov chain has no two disjoint closed sets. This fact is formally stated in the next theorem.

Theorem 1. *If an MDP is unichain, then Equation 1.3 has a solution.*

If Equation 1.3 has a solution, we can solve it by the iterative method in Equation (1.1). We restate Theorem 6.6.1 of (Tijms, 2003) in the theorem below, which allows us to use the recursion in Equation (1.1) to compute g and $h(\cdot)$.

Theorem 2. *For any state i, we have*

$$h(i) - h(0) = \lim_{n \to \infty} [v_n(i) - v_n(0)], \tag{1.5}$$

and

$$g = \lim_{n\to\infty} \frac{v_n(i)}{n}. \tag{1.6}$$

Furthermore,

$$\min_{i\in S}\{v_n(i) - v_{n-1}(i)\} \leq g \leq \max_{i\in S}\{v_n(i) - v_{n-1}(i)\} \tag{1.7}$$

However, solving the optimality equation is intractable when the state space and action space are large. Hence, we develop heuristic policies which perform well. One such policy is called the index policy, which we define below.

Suppose the MDP is unichain. Let π be a given initial policy. Then there exists a solution g^π and a bias function h^π that satisfy

$$h^\pi(i) + g^\pi = \min_{a\in\mathscr{A}}\{c(i,a) + \sum_{j\in S} p_{ij}(a)h^\pi(j)\}.$$

Now consider a policy $\hat{\pi}$ that chooses the action $\hat{a}$ in state i, where

$$\hat{a} \in \arg\min_a\{r(i,a) + \sum_{j\in S} p_{ij}(a)h^\pi(j)\}.$$

The next theorem shows the importance of this construction.

Theorem 3.

$$g^{\hat{\pi}} \leq g^{\pi}.$$

and if $g^{\hat{\pi}} = g^{\pi}$, then $\hat{\pi}$ is the average-cost optimal policy.

Suppose we can construct a function $f : S \times \mathscr{A} \to \mathbb{R}$ such that

$$\arg\min_a f(i,a) \subset \arg\min_a \{r(i,a) + \sum_{j \in S} p_{ij}(a) h^{\pi}(j)\}.$$

The function f is called an index function and the policy $\hat{\pi}$ is called an index policy using the index function f. It has been observed that the index policy $\hat{\pi}$ provides a tractable heuristic policy, especially if the initial policy π is chosen wisely.

We apply this methodology to two special problems in this dissertation. The rest of this dissertation is organized as follows. Chapter 2 describes the application of index policies in the appointment scheduling problem with patient preferences. Chapter 3 studies the application of index policies in the automated teller machine (ATM) replenishment problem. In each chapter we first formulate the problem as an MDP. Then we show the structural properties of the average-cost optimal policy. We derive an index policy and numerically study its performance.

CHAPTER 2: Appointment Scheduling with Patient Preference

2.1 Introduction

In recent years, more and more hospitals and clinics have started utilizing information technology, in particular the electronic medical record systems. This not only enables clinical staff (physicians, nurses, lab technicians and pharmacists) to provide high-quality service, but also allows the communication between patients and clinical staff to become increasingly smooth and seamless. The online appointment scheduling system is one of the manifestations of this recent development. For instance, ambulatory care patients are able to select their preferred appointment date, time and provider through eClinicalWorks Patient Portal, if their clinics have the appropriate software deployed. Another example is ZocDoc, which allows patients to register with an email address and helps them find doctors and book appointments through their website.

Despite the advanced technology, the patient preference has received limited attention in practice. Often the patients are not provided with many options when it comes to making appointments; see (Feldman et al., 2014). Many patients might prefer seeing the doctor as soon as possible while some other patients may like a later date due to their job constraints or the fact that the physician they usually see is not available at an earlier date. Taking their preferences into consideration and attempting to meet their needs will improve the service quality and the satisfaction of the patients; see (Feldman et al., 2014).

Clearly, one can meet the patients' preferences if enough resources are available to satisfy it. In practice this would mean overbooked schedules for the service providers and increased cost of service. Thus there is a trade-off between the level of patient satisfaction and the monetary cost to the service provider. We can measure the utility or dis-utility of the schedule by a cost function. We explain this in detail in Section 2.3.

The goal of the scheduling policy is to find this optimal trade-off between the level at which patients' preferences are satisfied and the cost of doing so. For example, one can aim to minimize the cost subject to the constraint that all patients' preferences must be satisfied. This leads to our base model described in Section 2.3. It is also possible to reduce the cost further if we allow the rejection of patients, which leads to an extension of the base model studied in Section 2.7.

We construct a model that helps clinics assign patients in the appointment schedule, taking both patients' preferences and the clinic's goals into consideration. We model the patient preferences by assuming that each patient has a set of appointment days that are equally acceptable to her. We consider a service provider which receives the appointment-booking requests one at a time and makes an appointment decision, while taking the currently scheduled appointments into account. In the base model, the patient is always given an appointment on one of the days of her choice. In the extended model of Section 2.7, the service provider may decide to reject the appointment request (and forgo any revenue from that patient). We also consider the possibility that patients who are scheduled to see a doctor may not actually show up for their appointments (that is, they are no-shows). Every patient that shows up produces a revenue for the clinic. The clinic typically overbooks the number of appointments to handle the no-shows. However, this implies that the clinic will frequently incur overtime costs, since the clinic is obligated to see all patients who arrive for their appointments on any given day. The objective is to minimize the long-run average cost by responding to the patients' booking requests based on their preferences.

We first analyze the problem using dynamic programming techniques. We identify the best policy that can be achieved theoretically and characterize the structure of the optimal policy. Since it is hard to implement the optimal policy, we consider several heuristic policies. Specifically, we introduce the shortest-queue policy, the randomized policy and also propose an index policy. We further show by numerical study that the index policy performs most closely to the optimal policy and is easy to implement.

The main contribution is in the novel model of patient preferences and that the appointment decisions are made after each arrival. A distinguishing feature of this model is that we can establish the unichain nature of the Markov decision process (MDP) and prove the existence of the average cost optimal policy.

The rest of Chapter 2 is organized as follows: we briefly review the related literature in Section 2.2. Details of the base model are described and assumptions are listed in Section 2.3. In Section 2.4 we study the optimal policy that minimizes the long-run average cost, and present the structure of the optimal policy. Section 2.5 introduces several heuristic polices, including our proposed index policy. The numerical study of these policies is given in Section 2.6. Section 2.7 describes the extension of the base model to incorporate the rejection option. We also briefly discuss how the heuristic policies of Section 2.5 can be modified to take into account the rejection option. Section 2.8 studies an appointment scheduling problem with a more-than-two-day horizon and general arrival processes. Finally, we conclude this chapter in Section 2.9.

2.2 Literature Review

This work belongs to a research area of clinical appointment scheduling problems in primary care setting. The literature in this area is quite extensive. We refer the reader to (Cayirli and Veral, 2003) and (Gupta and Denton, 2008), which provide excellent surveys of this area.

(Wang and Gupta, 2011) mention that most clinics use a two-step process to build the appointment scheduling system: “clinic profile setup” and “appointment booking”. The former deals with the problem of dividing the physician’s working time into appointment slots while the latter takes care of assigning the available slots to meet the requests from incoming patients. Many papers in this area are related to one of the two steps. Our work falls into the category of analyzing the problems arising in the second step.

An important aspect of appointment scheduling is the phenomenon of patient no-shows. (Green and Savin, 2008) and (Liu et al., 2010) report that the no-show rate depends on the length between the time a patient requests an appointment and the time she sees the physician. To deal with the effect of no-shows, a clinic might adopt the practice of overbooking. In this sense, (LaGanga and Lawrence, 2012) develop an effective, near-optimal solution procedure to solve this overbooking problem.

Many papers in the literature model and solve the appointment scheduling problem using a queueing theoretic approach. (Hassin and Mendel, 2008) determine the time intervals between the scheduled arrival times to minimize the patients' waiting cost and the physicians' availability cost. The authors incorporate no-shows and exponential service times in their model. (Kaandorp and Koole, 2007) choose the number of patients scheduled in each interval to minimize the weighted sum of patients' waiting time, doctor's idle time, and doctor's overtime. The authors develop a local search algorithm to search for the optimal schedule, assuming exponential service times. (Kuiper et al., 2015) determine the appointment times to minimize the quadratic loss function of patients' waiting time and physicians' idle time. The authors extend the distribution of service time to a more general form and propose computationally feasible approaches. (Zacharias and Pinedo, 2014) design the paradigms to assign the patients to the fixed appointment slots. The authors consider overbooking to counter the no-show effect and minimize the expected weighted sum of the patients' waiting times and the physician's idle time and overtime.

In the revenue management literature, there is a significant stream of articles regarding applying discrete choice models to analyze the consumer behavior, see for example (Talluri and Van Ryzin, 2004) for a detailed discussion. As far as we know, work on including the patient preferences in the appointment scheduling problem is limited and often involves stochastic dynamic programming; see (Balasubramanian et al., 2014; Gupta and Wang, 2008; Wang and Gupta, 2011; Feldman et al., 2014). (Balasubramanian et al., 2014) assume that the same-day patient accepts being assigned to the earliest available slot for a physician once the physician has been decided. (Gupta and Wang, 2008) consider that the patients have preferences over the slots instead of dates. They model the preferences via a discrete choice model, assuming that each patient has a preferred slot and the patient leaves if that slot is unavailable. Their assumption on the cost structure (the expected net profit is concave in the number of schedule appointments) is similar to ours. However, it is hard to obtain the patients choice probabilities for the clinics, because the front desk has no mechanism to collect such information. Furthermore, it is rare for the clinics to reveal the available slots to the patients. (Wang and Gupta, 2011) assume that all appointment requests are known at the beginning of a day and a clinic then decides upon the appointment schedule for the day, taking into account the random number of walkin patients. Our work explicitly allows multi-day appointments. (Feldman et al., 2014) model the patients preferences over the appointment dates using a discrete choice model. They assume that the clinic makes several appointment dates available for the patients to choose from. Our work differs from theirs in that we consider the patients revealing their preferences when they request the appointment.

2.3 Base Model

In this section, we propose our base model: patients arrive one at a time and request an appointment at a clinic. The clinic accepts appointment requests for the next two days, which is our scheduling horizon. There are no appointments given on the same day. The clinic's appointment book contains the number of scheduled appointments on the next two days (tomorrow and the day after tomorrow). We model the patient preference over the next two days as follows. Each patient has a set $\mathscr{A} \subset \{1, 2\}$ of the days that are acceptable to her. We call such a patient type-$\mathscr{A}$ patient. When she arrives to request an appointment, the clinic gives her an appointment t days into future for some $t \in \mathscr{A}$. On day n there are three types of patients: a type-1 patient only requesting an appointment on day $n + 1$; a type-2 patient only requesting an appointment on day $n+2$; a type-12 patient requesting an appointment on either day $n+1$ or day $n+2$. We denote the probability that an arriving patient belongs to each category by p_1, p_2 and p_{12}, where $p_1 + p_2 + p_{12} = 1$.

We observe the current system state (i, j), where i is the number of scheduled appointments on day $n+1$ and j is the number of scheduled appointments on day $n+2$. We then decide which day to schedule her appointment and update the system state accordingly. The arriving, observing, deciding and updating all happen instantaneously in that order.

Let A_n be the number of requests that arrive on day n. We assume the distribution of A_n is modified Geometric with parameter $1-\alpha$ (where $\alpha \in (0,1)$), that is, $\mathrm{P}(A_n = k) = \alpha^k(1-\alpha)$ where $k = 0, 1, 2, \ldots$. These are common assumptions in the previous literature; see (Zonderland et al., 2015). Throughout this chapter, we say a random variable having such a probability mass function follows a $\mathrm{G}(\alpha)$ distribution with expectation $\tau = \frac{\alpha}{1-\alpha}$ (τ requests arrive during each day on average). Because of the memoryless property of this distribution, the probability that an additional request arrives is α and with probability $1-\alpha$, there is no additional request coming in and day n is finished. The next day starts and this process of scheduling incoming appointment requests continues.

Let X_k be the state of the system just before the kth event for $k \geq 1$. (An event can be an arrival or a change of day.) The state is given by a pair of non-negative integers (i, j), where i is the number of appointments already scheduled for tomorrow and j is the number of appointments already scheduled for the day after tomorrow. Suppose $X_k = (i, j)$, and the kth event is an arrival. With probability p_1, p_2 or p_{12}, this arrival is a type-1, type-2 or type-12 patient. A decision D_k has to be made about accepting, rejecting, and scheduling the appointment for this arrival, and it will result in a change of state of the system from X_k to X_{k+1}. If the kth event is a change of day, then the current day is finished and the cost $c(i)$ is incurred, since i patients are scheduled for the next day. In this case the decision D_k is to do nothing. At the beginning of the next day, the system state changes to $(j, 0)$. This shows that $\{(X_k, D_k), k \geq 1\}$ is an MDP with the state space $S = \{(i, j) : i \geq 0, j \geq 0\}$.

A policy specifies a rule for selecting the decision D_k ($k \geq 1$). We say the policy is stationary Markovian if D_k depends only on X_k and not on k. A stationary Markovian policy assigns the type-12 patient on day n to day $n+1$ with probability $\phi(i, j)$ and to day $n+2$ with probability $1 - \phi(i, j)$, in state (i, j). Let Π denote the class of stationary Markovian policies, where each policy $\pi \in \Pi$ can be captured by the function $\phi : S \to [0, 1]$.

We now discuss the system dynamics after each event. Consider an arrival event. If the arrival is of type-1, the decision is to give her an appointment on the next day, then the state changes to $(i+1, j)$. If the arrival is of type-2, the decision is to give her an appointment on the day after next, then the state changes to $(i, j+1)$. If the arrival is of type-12, she can be assigned to either the next day, with state changing to $(i+1, j)$ or the day after next, with state changing to $(i, j+1)$. If the event is a change-of-day event, a cost $c(i)$ is incurred and the system state changes to $(j, 0)$. From the dynamics above, we get the optimality equation given below:

$$\begin{aligned} h(i,j) + g =& \alpha[p_{12} \min\{h(i+1,j), h(i,j+1)\} + p_1 h(i+1,j) \\ &+ p_2 h(i,j+1)] + (1-\alpha)[c(i) + h(j,0)]. \end{aligned} \tag{2.1}$$

Suppose there is a solution to Equation (2.1). Then g is the optimal long run average cost and $h(\cdot, \cdot)$ is the bias under the optimal policy. The policy π that chooses the action that minimizes the right hand side of Equation (2.1) provides the optimal decision for the arrival of type-12 patient.

We describe an iterative method to solve Equation (2.1). Set $v_0(i,j) = 0$ for all (i,j) and, for $k \geq 1$,

$$\begin{aligned} v_k(i,j) = \alpha[p_{12} \min\{v_{k-1}(i+1,j), v_{k-1}(i,j+1)\} + p_1 v_{k-1}(i+1,j) \\ + p_2 v_{k-1}(i,j+1)] + (1-\alpha)[c(i) + v_{k-1}(j,0)]. \end{aligned} \tag{2.2}$$

Note that one can interpret $v_k(i,j)$ as the total expected cost incurred over the first k events.

For a finite state space MDP, the existence of a solution in Equation (2.1) and the convergence of the value iteration method in Equation (2.2) has been well studied. However, for a countable state space MDP with unbounded cost (the category our MDP falls into), the results are limited. We use the Theorem 2.10 of (Blok and Spieksma, 2015) to show that Equation (2.1) has a solution and the value iteration method in Equation (2.2) can be used to compute it. The V-uniform geometric recurrence condition in Theorem 2.10 has been introduced and proved in (Dekker and Hordijk, 1992) and (Dekker et al., 1994). Both (Dekker et al., 1994) and (Spieksma, 1990) have shown this condition is equivalent with the V-uniform geometric ergodicity. We restate Theorem 2.10 of (Blok and Spieksma, 2015) below for a general MDP with a countable state space S, the transition probability $P^{\pi}_{x,y}$ from state x to state y under policy π, and the cost function $c(\cdot)$. We omit the continuity assumptions since they are automatically satisfied in our setting. We use the unichain assumption, that is, each stationary Markovian policy the associated Markov chain with a single closed sets; see (Tijms, 2003).

Theorem 4. *Suppose an MDP satisfies the following conditions.*

(a) There exists a function $V\colon S \to [1, \infty)$*, a finite set* $\mathbb{M} \subset S$ *and a constant* $\beta < 1$ *such that, for all* $\pi \in \Pi$

$$\sum_{y \notin \mathbb{M}} P^{\pi}_{x,y} V(y) \leq \beta V(x), \quad \forall x \in S. \tag{2.3}$$

(b) The cost $c(\cdot)$ *satisfies*

$$\sup_{x \in S} \frac{|c(x)|}{V(x)} < \infty.$$

(c) The MDP is unichain.

(d) The MDP is aperiodic.

Then, there exists a solution pair (g, h) *to Equation* (2.1)*, and* $h(\cdot, \cdot)$ *is given by*

$$h(i, j) - h(0, 0) = \lim_{k \to \infty} [v_k(i, j) - v_k(0, 0)].$$

We show that the conditions of Theorem 4 are satisfied by our MDP.

Theorem 5. *Suppose* a*,* b*, and* M *satisfy*

$$1 < a < b < \frac{1}{\alpha}, \quad \left(\frac{a}{b}\right)^M < \frac{1 - \alpha b}{1 - \alpha}. \tag{2.4}$$

Let $\mathbb{M} = \{(i,j) : 0 \leq i+j \leq M\} \subset S$ be the finite set and define $V(i,j) = a^i b^j$. Furthermore, suppose $c(i)$ is bounded by a polynomial in i. Then all the conditions in Theorem 4 are satisfied.

Proof. (a) It suffices to find a constant $\beta < 1$ which satisfies Equation (2.3) for all $\pi \in \Pi$. Consider the policy π that assigns the type-12 patient arriving on day n to day $n+1$ with probability $\phi(i,j)$ and to day $n+2$ with probability $1-\phi(i,j)$. Given the current state $x = (i,j)$, the next state y is given by

$$y = \begin{cases} (i+1,j), & \text{w.p. } \alpha(p_1 + p_{12}\phi(i,j)); \\ (i,j+1), & \text{w.p. } \alpha(p_2 + p_{12}(1-\phi(i,j))); \\ (j,0), & \text{w.p. } 1-\alpha. \end{cases}$$

(i) For state (i,j) where $j \leq M, i+j \leq M-1$, any $\beta \in (0,1)$ makes Equation (2.3) hold since the left-hand side is 0.

(ii) For state (i,j) where $j \leq M, i+j \geq M$, we need to find $\beta < 1$ such that

$$\alpha(p_1 + p_{12}\phi(i,j))V(i+1,j) + \alpha(p_2 + p_{12}(1-\phi(i,j)))V(i,j+1) \leq \beta V(i,j)$$

holds. Using $V(i,j) = a^i b^j$ we get

$$\alpha(p_1 + p_{12}\phi(i,j))a^{i+1}b^j + \alpha(p_2 + p_{12}(1-\phi(i,j)))a^i b^{j+1} \leq \beta a^i b^j,$$

which reduces to

$$\alpha(p_1 + p_{12}\phi(i,j))a + \alpha(p_2 + p_{12}(1 - \phi(i,j)))b \leq \beta.$$

Since $a < b$, we have

$$\alpha(p_1 + p_{12}\phi(i,j))a + \alpha(p_2 + p_{12}(1 - \phi(i,j)))b \leq \alpha b,$$

so that we choose $\beta = \alpha b$. Since $b < \frac{1}{\alpha}$, we see that $\beta < 1$.

(iii) For state (i,j) where $j > M$, we need to find $\beta < 1$ such that

$$\alpha(p_1 + p_{12}\phi(i,j))V(i+1,j) + \alpha(p_2 + p_{12}(1 - \phi(i,j)))V(i,j+1) + (1-\alpha)V(j,0)$$
$$\leq \beta V(i,j).$$

Using $V(i,j) = a^i b^j$ we get

$$\alpha(p_1 + p_{12}\phi(i,j))a^{i+1}b^j + \alpha(p_2 + p_{12}(1 - \phi(i,j)))a^i b^{j+1} + (1-\alpha)a^j b^0 \leq \beta a^i b^j.$$

This reduces to

$$\alpha(p_1 + p_{12}\phi(i,j))a + \alpha(p_2 + p_{12}(1 - \phi(i,j)))b + (1-\alpha)\frac{a^j}{a^i b^j} \leq \beta.$$

Since $a < b$, it suffices to choose $\beta < 1$ such that

$$\alpha b + (1-\alpha)\frac{a^j}{a^i b^j} \le \beta.$$

Since $1 < a < b$, we get

$$\frac{a^j}{a^i b^j} < \frac{a^j}{b^j} < \left(\frac{a}{b}\right)^M.$$

We choose

$$\beta = \alpha b + (1-\alpha)\left(\frac{a}{b}\right)^M. \tag{2.5}$$

We know $\beta < 1$ since

$$\left(\frac{a}{b}\right)^M < \frac{1-\alpha b}{1-\alpha}.$$

The β in Equation (2.5) satisfies Equation (2.3). This proves part (a) of Theorem 4.

(b) Since $c(i)$ is bounded by a polynomial of i, we have

$$\sup_{(i,j)\in S} \frac{|c(i)|}{a^i b^j} < \infty.$$

(c) We have, for any policy π, $P(X_{k+1} = (j,0)|X_k = (i,j)) = 1-\alpha, P(X_{k+2} = (0,0)|X_{k+1} = (j,0)) = 1-\alpha$. The system can go from any state (i,j) to state $(0,0)$ in two steps with a positive probability, regardless of the policy followed. Therefore $\{(X_k, D_k), k \geq$

$1\}$ satisfies the unichain assumption.

(d) The state $(0,0)$ is aperiodic, because $\mathrm{P}(X_{k+1} = (0,0)|X_k = (0,0)) = 1 - \alpha$. The MDP is aperiodic since it satisfies the unichain assumption. $\square$

Remark: A triplet (a, b, M) satisfying Equation (2.4) is given by:

$$(a, b, M) = \left(1 + \frac{1}{3}\left(\frac{1}{\alpha} - 1\right), 1 + \frac{2}{3}\left(\frac{1}{\alpha} - 1\right), \left\lceil \frac{\log(1-\alpha) - \log(1-\alpha b)}{\log b - \log a} \right\rceil + 1\right).$$

Theorem 5 enables us to use the recursions in Equation (2.2) to compute the optimal average cost g and the bias $h(\cdot, \cdot)$, which can be used to derive the optimal policy. Note that g is the long-run average cost per event, so that the long-run average cost per day is given by $\frac{g}{1-\alpha}$.

Let $d_t(i, j)$ be the optimal decision when a type-t ($t = 1, 2, 12$) patient arrives on day n and sees the system in state (i, j). We write $d_t(i, j) = 1$ if the decision is to assign the patient to day $n + 1$, $d_t(i, j) = 2$ if the decision is to assign the patient to day $n + 2$. Since we must assign type-1 and type-2 patient to their desired day, $d_1(i, j) = 1$ and $d_2(i, j) = 2$. The standard theory of dynamic programming (see (Tijms, 2003)) shows that the optimal policy for type-12 patients can be computed from the bias $h(\cdot, \cdot)$ as follows:

$$d_{12}(i, j) = \begin{cases} 1, \text{ if } h(i+1, j) \leq h(i, j+1) \\ 2, \text{ if } h(i+1, j) > h(i, j+1). \end{cases} \tag{2.6}$$

2.4 Structural Properties of Optimal Policy

In this section, we study the structural properties of the optimal policy in the base model and then use numerical computations to illustrate them. Theorem 6 below gives the structural properties of the bias $h(\cdot, \cdot)$ of Equation (2.1). We use the event-based dynamic programming (DP) techniques of (Koole, 2007) to prove the following structural properties.

Let $f : S \to \mathbb{R}$, then we define:

$$Convexity: \begin{cases} f(i,j) + f(i+2,j) \geq 2f(i+1,j), \quad \forall (i,j) \in S; \\ f(i,j) + f(i,j+2) \geq 2f(i,j+1), \quad \forall (i,j) \in S. \end{cases}$$

$$Super: \ f(i,j) + f(i+1,j+1) \geq f(i+1,j) + f(i,j+1), \quad \forall (i,j) \in S.$$

$$SuperC: \begin{cases} f(i+2,j) + f(i,j+1) \geq f(i+1,j) + f(i+1,j+1), \quad \forall (i,j) \in S; \\ f(i+1,j) + f(i,j+2) \geq f(i,j+1) + f(i+1,j+1), \quad \forall (i,j) \in S. \end{cases}$$

Next, we define some useful operators

$$T_{12}f(i,j) = \min\{f(i+1,j), f(i,j+1)\}.$$

$$T_1 f(i,j) = f(i+1,j).$$

$$T_2 f(i,j) = f(i,j+1).$$

$$T_p(f_{12}, f_1, f_2)(i,j) = p_{12}f_{12}(i,j) + p_1 f_1(i,j) + p_2 f_2(i,j).$$

$$T_\alpha(f_1, f_2)(i,j) = \alpha f_1(i,j) + (1-\alpha) f_2(i,j).$$

$$T_c f(i,j) = c(i) + f(j,0).$$

so that Equation (2.2) can be written as

$$v_{k+1} = T_\alpha(T_p(T_{12}v_k, T_1v_k, T_2v_k), T_cv_k), \tag{2.7}$$

with $v_0(i,j) = 0$, for all $(i,j) \in S$. We adopt the notations from (Koole, 1998): $T : P_1, ..., P_k \to P_1$ means for operator T, if f has the properties $P_1, ..., P_k$ then Tf has the property P_1.

Lemma 1. *The operators T_1, T_2, T_p, T_α satisfy:*

(a) $Convexity \to Convexity$;

(b) $Super \to Super$;

(c) $SuperC \to SuperC$.

Proof. For operators T_1 and T_2, the result is trivial. For operators T_p and T_α, the results follow from the fact that $Convexity, Super, SuperC$ are closed under convex combination. □

Lemma 2. *Assume $c(\cdot)$ is convex. Then the operators T_{12} and T_c satisfy:*

(a) $Convexity, Super, SuperC \to Convexity$;

(b) $Convexity, Super, SuperC \to Super$;

(c) $Convexity, Super, SuperC \to SuperC$.

Proof. The operator T_{12} coincides with the $T_{R(I)}$ in (Koole, 1998), which yields the results. For the operator T_c, the $Convexity$ of T_cf follows from the $Convexity$ of f. The $Super$ of

$T_c f$ is trivially satisfied. For $SuperC$ of $T_c f$, we have

$$\begin{aligned}
T_c f(i+2,j) + T_c f(i,j+1) &= c(i+2) + f(j,0) + c(i) + f(j+1,0) \\
&\geq c(i+1) + f(j,0) + c(i+1) + f(j+1,0) \\
&= T_c f(i+1,j) + T_c f(i+1,j+1).
\end{aligned}$$

Similarly,

$$\begin{aligned}
T_c f(i+1,j) + T_c f(i,j+2) &= c(i+1) + f(j,0) + c(i) + f(j+2,0) \\
&\geq c(i) + f(j+1,0) + c(i+1) + f(j+1,0) \\
&= T_c f(i,j+1) + T_c f(i+1,j+1).
\end{aligned}$$

This yields the $SuperC$ of $T_c f$. □

Theorem 6. *Assume $c(\cdot)$ is convex. The bias $h(\cdot,\cdot)$ satisfying Equation* (2.1) *has the following properties: $Convexity, Super, SuperC$.*

Proof. By Lemma 1 and 2, the induction hypothesis leads to the fact that the functions v_k in Equation (2.7) satisfy these properties. By Theorems 4 and 5, the bias $h(\cdot,\cdot)$ has these properties. □

The next corollary gives the structure of the optimal decision d_{12}.

Corollary 1. $d_{12}(i,j) = 1 \Rightarrow d_{12}(i,j+1) = 1,\ \ d_{12}(i,j) = 2 \Rightarrow d_{12}(i+1,j) = 2,\ \ \forall (i,j) \in S$.

Proof. The $SuperC$ of $h(\cdot,\cdot)$ gives rise to the fact that $h(i+1,j)-h(i,j+1)$ increases with i for any fixed j, and $h(i+1,j)-h(i,j+1)$ decreases with j for any fixed i. We have

$$h(i+1,j) \leq h(i,j+1) \Rightarrow h(i+1,j+1) \leq h(i,j+2),$$

$$h(i+1,j) > h(i,j+1) \Rightarrow h(i+2,j) > h(i+1,j+1).$$

Combining this with the definition of $d_{12}(i,j)$ in Equation (2.6) we have the results. □

Theorem 6 and Corollary 1 yield the following structure of the optimal policy.

Theorem 7. *Assume $c(\cdot)$ is convex. There exists a critical number $j^*(i)$ for each i such that*

$$d_{12}(i,j) = \begin{cases} 1, & \textit{if } j \geq j^*(i); \\ 2, & \textit{if } j < j^*(i). \end{cases}$$

Furthermore, there exists a critical number $i^*(j)$ for each j such that

$$d_{12}(i,j) = \begin{cases} 1, & \textit{if } i < i^*(j); \\ 2, & \textit{if } i \geq i^*(j). \end{cases}$$

2.4.1 Numerical Illustration

We now present a numerical illustration of the optimal policies under a specific $c(\cdot)$ function. Suppose a patient shows up for her appointment with probability p, and is a no-show with probability $1-p$. We also assume that the clinic collects revenue r from each patient served. There is no penalty for no-show patients. The fixed cost of the clinic is K, for serving up to m_0 patients. When serving more than m_0 patients, the variable cost of each patient served is $c_o > r$, say overtime or overstaffing cost. The net cost $F(y)$ when there are y patients who actually get served by the clinic during one day is

$$F(y) = K - ry + c_o \max\{y - m_0, 0\}. \tag{2.8}$$

The net cost incurred on a day with x scheduled appointments is given by $c(x) = \mathrm{E}[F(\mathrm{Bin}(x,p))]$, where $\mathrm{Bin}(x,p)$ is a Binomial random variable with parameters x and p, representing the number of patients who actually show up for their appointments. One can show that $c(\cdot)$ is a convex function, which takes its minimum at $m \geq m_0$. For example, if we choose parameters $K = 440, r = 40, c_o = 50, m_0 = 11, p = 0.8$, the $c(\cdot)$ function is minimized at $m = 15$, which is depicted in Figure 2.1. We choose $\alpha = \frac{15}{16}$, so that the expected number of patients arriving each day is 15, which is also the value of x at which $c(x)$ is minimized.

Next we discuss how to compute $v_k(i,j)$ in the value iteration method. First we truncate the entire state space $[0,\infty) \times [0,\infty)$ to $[0,T] \times [0,T]$ in numerical calculations. The calculations of $v_{k+1}(i,T)$ and $v_{k+1}(T,j)$ involve $v_k(i,T+1)$ and $v_k(T+1,j)$, which are not computed in the numerical program. Therefore, we use the following approximations

$$v_{k+1}(i,T+1) \approx v_k(i,T) + [v_k(i,T) - v_k(i,T-1)], \;\; 0 \leq i \leq T.$$

$$v_{k+1}(T+1,j) \approx v_k(T,j) + [v_k(T,j) - v_k(T-1,j)], \;\; 0 \leq j \leq T.$$

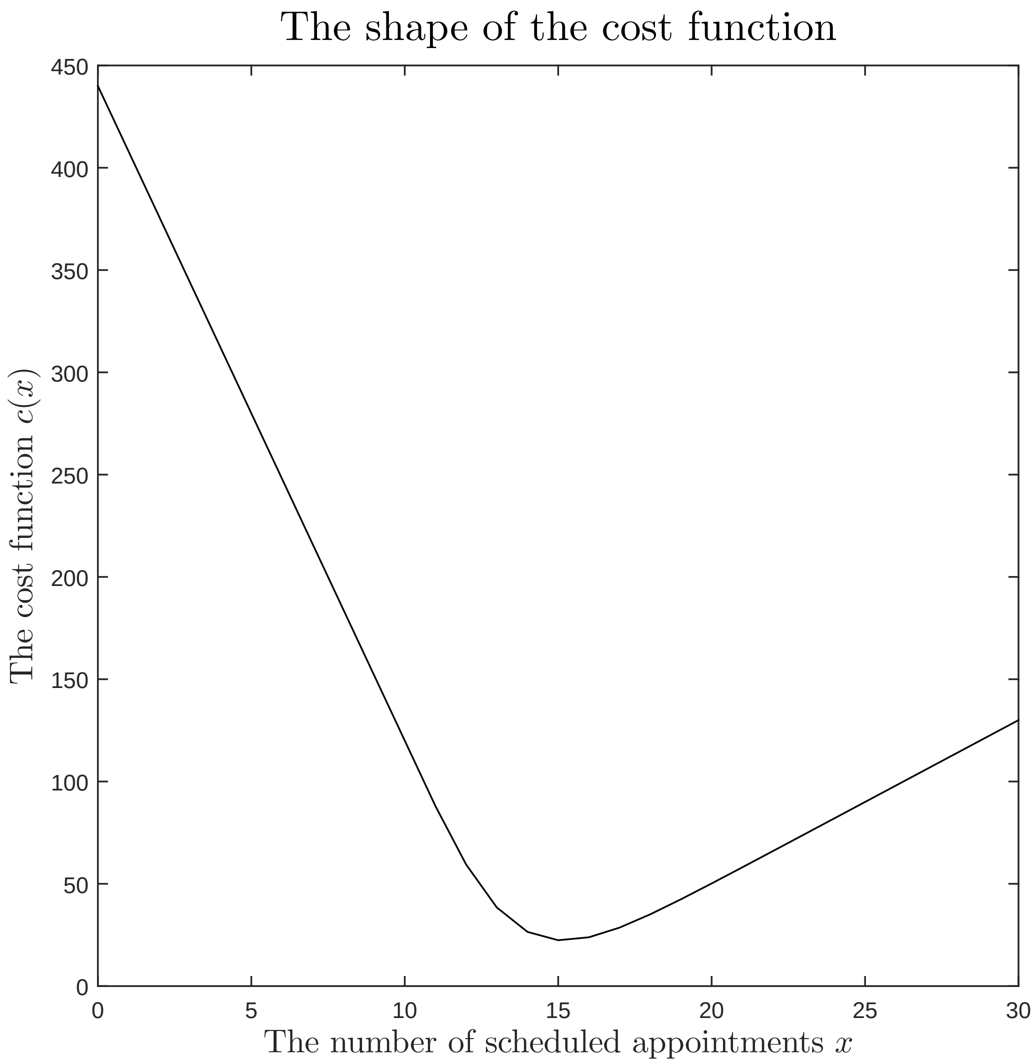

Figure 2.1: The cost function $c(x) = \mathrm{E}[F(\mathrm{Bin}(x, p))]$ with F as defined in Equation (2.8).

Here, we use the Taylor's approximation $v(x+1) \approx v(x) + v'(x)$ and use $v'(x) \approx v(x) - v(x-1)$ to approximate the derivative. This approximation exploits the convexity of $v_k(i, j)$ for fixed i or j; see (Ha, 1997) for a similar use of this approximation. We use the recursion in Equation (2.2) to calculate the long-run average cost under the optimal policy. From (Tijms, 2003), we know that

$$\min_{i,j}\{v_k(i,j) - v_{k-1}(i,j)\} \leq g \leq \max_{i,j}\{v_k(i,j) - v_{k-1}(i,j)\},$$

so we use a stopping criterion

$$\max_{i,j}\{v_k(i,j) - v_{k-1}(i,j)\} - \min_{i,j}\{v_k(i,j) - v_{k-1}(i,j)\} \leq \epsilon,$$

where we set ϵ to 0.01. We observed that the relative error is less than 0.1% of the optimal value when the algorithm stops, using $T = 120$ throughout our numerical study. Note that given the expected number of appointment requests on each day is 15, the probability that there are more than 120 scheduled patients is less than 0.0005.

Figure 2.2 displays the optimal decision for a type-12 patient in each state under the parameters $\alpha = \frac{15}{16}, p_1 = 0.1, p_2 = 0.1, p_{12} = 0.8$. Figure 2.2 clearly shows a switching-curve pattern, and in Table 2.1 the corresponding values of $j^*(i)$ and $i^*(j)$ are given up to 15 appointments.

Remark: In theory it is possible that $i^*(j)$ and $j^*(i)$ are equal to zero or infinity. However, over all our experiments we have observed that only $j^*(i)$ attained zero.

Table 2.1: $j^*(i)$ and $i^*(j)$ with parameters $\alpha = \frac{15}{16}, p_1 = 0.1, p_2 = 0.1, p_{12} = 0.8$.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$j^*(i)$	0	0	0	0	0	0	0	0	0	0	0	0	5	11	13	14
j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$i^*(j)$	12	12	12	12	12	13	13	13	13	13	13	14	14	15	16	16

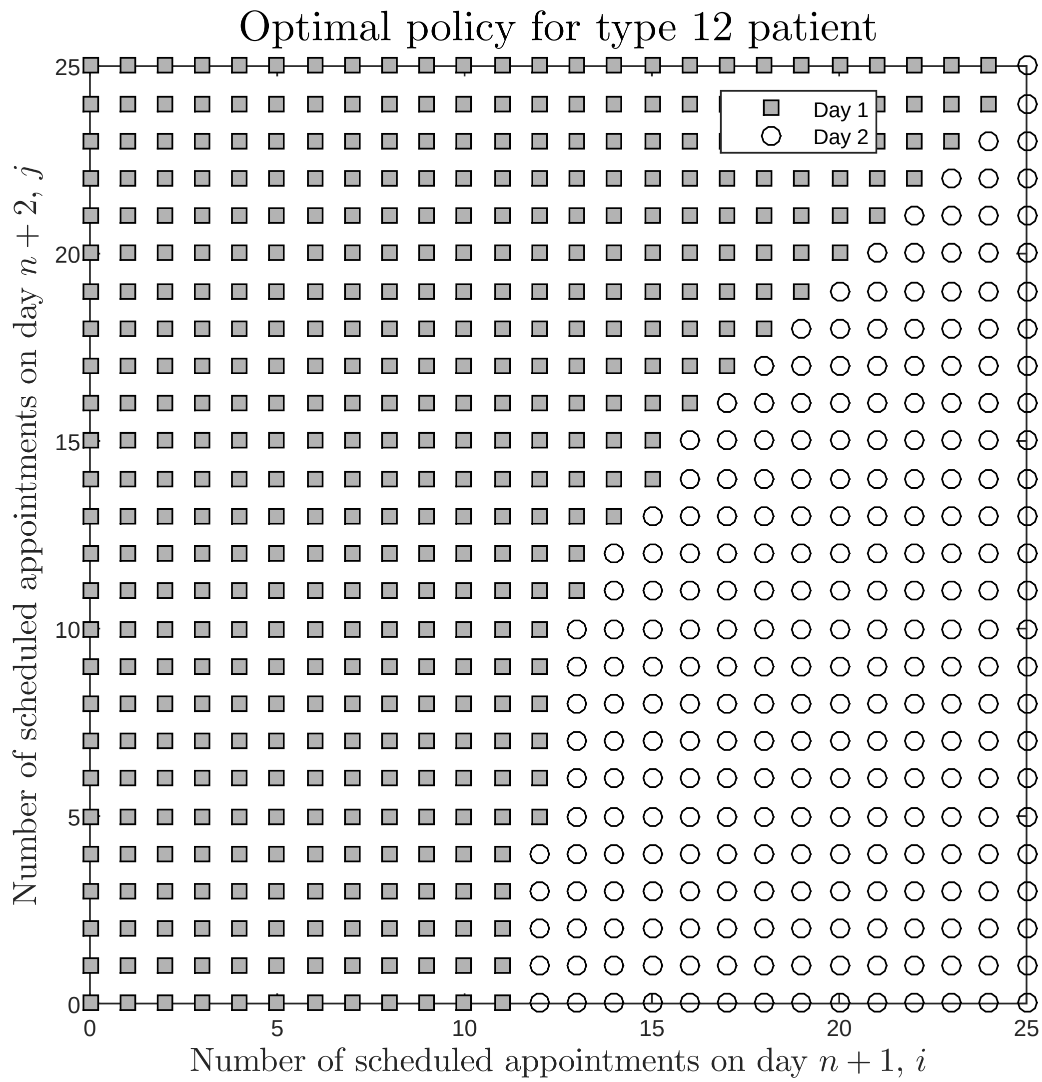

Figure 2.2: The optimal policy for type-12 patients.

2.5 Heuristic Policies

In general the optimal policy is intractable to implement, since its structure (given by the switching curve $i^*(j)$) is not available in an analytical form. Therefore we study three intuitive policies, which are described by the decision to be taken when a type-12 patient arrives.

2.5.1 Shortest-Queue Policy (SQ)

Under the shortest-queue (SQ) policy we assign a type-12 patient to the "shortest" queue, i.e., we assign her on the day with the fewest scheduled appointments. The appointment decision $d^{SQ}(i, j)$ by the SQ policy in state (i, j) is given by

$$d_{12}^{SQ}(i,j) = \begin{cases} 1, & \text{if } i < j; \\ 2, & \text{if } i > j. \end{cases}$$

When the numbers of scheduled appointments are equal on both days, we assign the patient to either day with probability 0.5.

The shortest-queue policy is equivalent to a myopic policy, under which the decisions are made ignoring cost incurred in the future. Under the myopic policy, the decision to give a type-12 patient arriving on day n and seeing the state (i, j) an appointment on day $n + 1$ incurs the cost $c(i+1)+c(j)$ while assigning her on day $n+2$ incurs the cost $c(i)+c(j+1)$. The convexity of the cost function $c(\cdot)$ implies the equivalence of the myopic policy and the shortest-queue policy. In mathematical terms, $c(i + 1) + c(j) < c(i) + c(j + 1) \Leftrightarrow c(i + 1) - c(i) < c(j + 1) - c(j) \Leftrightarrow i < j$.

2.5.2 Randomized Policy (RP)

The randomized policy (RP) assigns a type-12 patient arriving on day n in state (i, j) to day $n + 1$ with probability θ and to day $n + 2$ with probability $1 - \theta$. The appointment decision $d^{RP}(i,j)$ by the RP in state (i,j) is given by

$$d_{12}^{RP}(i,j) = \begin{cases} 1, & \text{with probability } \theta; \\ 2, & \text{with probability } 1-\theta. \end{cases}$$

Let ρ_t ($t = 1, 2$) be the fractions of incoming patients that get assigned to day $n+t$ under RP. We see that

$$\rho_1 = p_1 + \theta p_{12}, \quad \rho_2 = p_2 + (1-\theta)p_{12}. \tag{2.9}$$

We say that $\theta \in [0,1]$ balances the number of appointments on each day if ρ_1 and ρ_2 solve the optimization problem below:

$$\begin{aligned} \text{min} \;\; & (\rho_1 - 0.5)^2 + (\rho_2 - 0.5)^2 \\ \text{s.t.} \;\; & p_1 \leq \rho_1 \leq p_1 + p_{12}, \\ & p_2 \leq \rho_2 \leq p_2 + p_{12}, \\ & \rho_1 + \rho_2 = 1. \end{aligned}$$

The constraints in the optimization problem ensure that there is a balancing $\theta \in [0,1]$ that will achieve the optimal ρ_1 and ρ_2 produced by the above optimization problem. Proposition 1 gives the choice of this θ.

Proposition 1. *The following θ balances the number of appointments on each day.*

$$\theta = \begin{cases} 0, & \textit{if } p_1 \geq 0.5; \\ 1, & \textit{if } p_2 \geq 0.5; \\ \frac{0.5-p_1}{p_{12}}, & \textit{otherwise}. \end{cases} \tag{2.10}$$

Proof. This optimization problem is equivalent to:

$$\begin{aligned} &\min \;\; (\rho_1 - 0.5)^2 \\ &\text{s.t.} \;\; p_1 \leq \rho_1 \leq p_1 + p_{12}, \end{aligned}$$

which has the solution

$$\rho_1 = \begin{cases} p_1, & \text{if } p_1 \geq 0.5; \\ 1 - p_2, & \text{if } p_2 \geq 0.5; \\ 0.5, & \text{otherwise}. \end{cases}$$

This is equivalent to the θ given in Equation (2.10). □

2.5.3 Index Policy

Now we develop a heuristic policy, called the index policy (IP), under which the decision made depends on the indices that are functions of the system state and the known parameters. Specifically we compute two index functions $\mathrm{I}_1(\cdot)$ and $\mathrm{I}_2(\cdot)$ and define the decision $d_{12}^{IP}(i, j)$

as follows:

$$d_{12}^{IP}(i,j) = \begin{cases} 1, & \text{if } \mathrm{I}_1(i) \le \mathrm{I}_2(j); \\ 2, & \text{if } \mathrm{I}_1(i) > \mathrm{I}_2(j). \end{cases} \tag{2.11}$$

Structurally it is the same as that of the optimal policy and this is the reason why we expect that the index policy performs well. We will derive explicit expressions for the index functions I_1 and I_2 which makes the implementation of this policy tractable.

We apply the one-step policy improvement algorithm to derive these indices. Consider a policy π_t ($t = 1, 2$) that assigns a type-12 patient arriving on day n to day $n + t$ but switches to the randomized policy from the next patient on.

Let γ denote the randomized policy. Its bias h_γ satisfies

$$\begin{aligned} h_\gamma(i,j) + g_\gamma = \alpha p_{12}[\theta h_\gamma(i+1,j) + (1-\theta)h_\gamma(i,j+1)] + \alpha p_1 h_\gamma(i+1,j) \\ + \alpha p_2 h_\gamma(i,j+1) + (1-\alpha)[c(i) + h_\gamma(j,0)], \end{aligned}$$

where g_γ is the long-run average cost under policy γ starting from state (i, j). If we perform a one-step improvement of the randomized policy, we see that in state (i, j) the improved policy assigns the patient to day $n + 1$ if $h_\gamma(i + 1, j) \le h_\gamma(i, j + 1)$ and to day $n + 2$ otherwise. We show that $h_\gamma(i+1, j) - h_\gamma(i, j)$ is independent of j and $h_\gamma(i, j+1) - h_\gamma(i, j)$ is independent of i, so we are able to define the indices as $\mathrm{I}_1(i) = h_\gamma(i + 1, j) - h_\gamma(i, j)$ and $\mathrm{I}_2(j) = h_\gamma(i, j+1) - h_\gamma(i, j)$. Although $h_\gamma(\cdot, \cdot)$ is hard to obtain, we see that its differences, namely $\mathrm{I}_1(i)$ and $\mathrm{I}_2(j)$, are easier to compute.

Suppose the state of the system is (i, j) and the policy γ is followed. For simplicity, we use A_n to denote the number of remaining patients who arrive on day n. We know A_n remains a $\mathrm{G}(\alpha)$ random variable due to the memoryless property. Of these A_n patients, the randomized policy assigns $A_{n,i}$ to day $n + i$. By the end of day n, the system state is $(i + A_{n,1}, j + A_{n,2})$. At the beginning of day $n + 1$, cost $c(i + B_1)$ is incurred where $B_1 = A_{n,1}$ and the system state is updated to $(j + A_{n,2}, 0)$. Following a similar argument, by the end of the day $n + 1$, the system state becomes $(j + A_{n,2} + A_{n+1,1}, A_{n+1,2})$. Then, at the beginning of day $n + 2$, we incur cost $c(j + B_2)$, where $B_2 = A_{n,2} + A_{n+1,1}$, and the system state changes to $(A_{n+1,2}, 0)$. Therefore the bias $h_\gamma(i, j)$ of following policy γ starting from state (i, j), satisfies the equation

$$h_\gamma(i, j) = \mathrm{E}[c(i + B_1)] + \mathrm{E}[c(j + B_2)] + \mathrm{E}[h_\gamma(A_{n+1,2}, 0)].$$

This implies that the effect of the initial state (i, j) disappears from the third day on. The index functions are given as:

$$\mathrm{I}_1(i) = h_\gamma(i + 1, j) - h_\gamma(i, j) = \mathrm{E}[c(i + 1 + B_1) - c(i + B_1)],\ i \geq 0;$$

$$\mathrm{I}_2(j) = h_\gamma(i, j + 1) - h_\gamma(i, j) = \mathrm{E}[c(j + 1 + B_2) - c(j + B_2)],\ j \geq 0.$$

Remark: Because the distributions of B_1, B_2 do not depend on i, j, the convexity of $c(\cdot)$ implies that $\mathrm{I}_1(i)$ and $\mathrm{I}_2(j)$ are increasing functions in i, j. The decisions under the index policy, namely $d_{12}^{IP}(i,j)$, in Equation (2.11) can be written as

$$d_{12}^{IP}(i,j) = \begin{cases} 1, & \text{if } i \leq \mathrm{I}_1^{-1}(\mathrm{I}_2(j)); \\ \\ 2, & \text{if } i > \mathrm{I}_1^{-1}(\mathrm{I}_2(j)). \end{cases}$$

The next theorem gives the distributions of $A_{n,1}, A_{n,2}$ and B_1, B_2.

Theorem 8. *Let A_n be a $\mathrm{G}(\alpha)$ random variable, and suppose each arriving patient is assigned to day $n+i$ with probability ρ_i. Let $\alpha_i = \frac{\alpha\rho_i}{1-\alpha(1-\rho_i)}$. Then $A_{n,i}$ is a $\mathrm{G}(\alpha_i)$ random variable. Furthermore*

$$\mathrm{P}(B_1 = k) = \alpha_1^k(1-\alpha_1), \;\; \mathrm{P}(B_2 = k) = \frac{\alpha_1^{k+1} - \alpha_2^{k+1}}{\alpha_1 - \alpha_2}(1-\alpha_1)(1-\alpha_2).$$

Proof. Let $\{Y_n, n \geq 1\}$ be i.i.d. Bernoulli(ρ) random variables and $X \sim \mathrm{G}(\alpha)$, then

$$Z = \sum_{n=1}^{X} Y_n \sim \mathbf{G}\left(\frac{\alpha\rho}{1-\alpha(1-\rho)}\right).$$

We know $A_n \sim \mathrm{G}(\alpha)$. For each patient, the probability that she is scheduled on day $n+i$ is ρ_i. Applying the result above, we know the number of patients who are scheduled on day $n+i$ follows the distribution $\mathrm{G}(\cdot)$, i.e.,

$$A_{n,i} \sim \mathrm{G}\left(\frac{\alpha\rho_i}{1-\alpha(1-\rho_i)}\right), \;\; i = 1, 2.$$

Because $B_1 = A_{n,1}, B_2 = A_{n,2} + A_{n+1,1}$ and $A_{n,2}$ and $A_{n+1,1}$ are independent, the theorem follows. $\square$

Using the above distributions of $A_{n,1}, A_{n,2}$ and B_1, B_2, we compute the indices. First, we write the function F of Equation (2.8) in the following form

$$F(y) = \begin{cases} r_1 y + b_1, & \text{if } y < m_0; \\ r_2 y + b_2, & \text{if } y \geq m_0, \end{cases}$$

where $r_1 = -r, b_1 = K, r_2 = c_o - r, b_2 = K - c_o m_0$. Second, the cost incurred at the beginning of a day is $c(x) = \mathrm{E}[F(\mathrm{Bin}(x,p))]$ where x is the number of scheduled appointments on that day and p is the show-up probability. The main result follows.

Theorem 9. *Let p, r_1, r_2 be as given above, ρ_i and α_i be as those in Theorem 8 and $\beta_i = \frac{\alpha_i p}{1-\alpha_i(1-p)}$, for $i = 1, 2$. The indices are given by*

$$\mathrm{I}_1(i) = r_1 p + p(r_2 - r_1)\mathrm{P}(\mathrm{Bin}(i + B_1, p) \geq m_0),\ i \geq 0;$$

$$\mathrm{I}_2(j) = r_1 p + p(r_2 - r_1)\mathrm{P}(\mathrm{Bin}(j + B_2, p) \geq m_0),\ j \geq 0,$$

where $\mathrm{P}(\mathrm{Bin}(i + B_1, p) \geq m_0) =$

$$\begin{cases} \beta_1^{m_0} (\frac{1}{\alpha_1})^i, & \textit{if } i \leq m_0, \\ \sum_{l=0}^{m_0} \binom{i}{l} p^l (1-p)^{i-l} \cdot \beta_1^{m_0 - l} + \sum_{l=m_0+1}^{i} \binom{i}{l} p^l (1-p)^{i-l}, & \textit{if } i > m_0, \end{cases}$$

and for $j \leq m_0$*,*

$$\mathrm{P}(\mathrm{Bin}(j+B_2,p) \geq m_0) = \frac{p^{m_0}}{\alpha_1-\alpha_2}\left[\frac{(1-\alpha_2)\alpha_1^{m_0+1-j}}{(1-\alpha_1+\alpha_1 p)^{m_0}} - \frac{(1-\alpha_1)\alpha_2^{m_0+1-j}}{(1-\alpha_2+\alpha_2 p)^{m_0}}\right]$$

and for $j > m_0$*,*

$$\begin{aligned}\mathrm{P}(\mathrm{Bin}(j+B_2,p) \geq m_0) = &\sum_{l=0}^{m_0}\binom{j}{l}p^l(1-p)^{j-l}\frac{(1-\alpha_1)(1-\alpha_2)}{p(\alpha_1-\alpha_2)}\left[\frac{\beta_1^{m_0-l+1}}{1-\beta_1} - \frac{\beta_2^{m_0-l+1}}{1-\beta_2}\right]\\ &+ \sum_{l=m_0+1}^{j}\binom{j}{l}p^l(1-p)^{j-l}.\end{aligned}$$

Proof. In order to calculate the indices, we first use a “sample-path” technique to compute

$$\begin{aligned}\mathrm{E}[c(x+1)-c(x)] &= \mathrm{E}[F(\mathrm{Bin}(x+1,p)] - \mathrm{E}[F(\mathrm{Bin}(x,p)]\\ &= \mathrm{E}[F(\mathrm{Bin}(x,p)+\mathrm{Bernoulli}(p)) - F(\mathrm{Bin}(x,p))]\\ &= p\mathrm{E}[F(\mathrm{Bin}(x,p)+1) - F(\mathrm{Bin}(x,p))]\\ &= r_1 p\mathrm{P}(\mathrm{Bin}(x,p) < m_0) + r_2 p\mathrm{P}(\mathrm{Bin}(x,p) \geq m_0)\\ &= r_1 p(1-\mathrm{P}(\mathrm{Bin}(x,p) \geq m_0)) + r_2 p\mathrm{P}(\mathrm{Bin}(x,p) \geq m_0)\\ &= r_1 p + p(r_2-r_1)\mathrm{P}(\mathrm{Bin}(x,p) \geq m_0).\end{aligned}$$

Therefore the indices are

$$\mathrm{I}_1(i) = \mathrm{E}[c(i+1+B_1) - c(i+B_1)] = r_1 p + p(r_2-r_1)\mathrm{P}(\mathrm{Bin}(i+B_1,p) \geq m_0);$$

$$\mathrm{I}_2(j) = \mathrm{E}[c(j+1+B_2) - c(j+B_2)] = r_1 p + p(r_2-r_1)\mathrm{P}(\mathrm{Bin}(j+B_2,p) \geq m_0).$$

The computation of $\mathrm{P}(\mathrm{Bin}(i+B_1,p) \geq m_0)$ and $\mathrm{P}(\mathrm{Bin}(j+B_2,p) \geq m_0)$ is straightforward.

$\square$

2.6 Numerical Study

In this section, we compare the performances of the heuristic policies with that of the optimal policy under the long-run average cost criterion. Before we do so, we discuss two metrics.

The most commonly used metric in the literature to compare policies is the percentage improvement in the cost of the optimal policy OP over the given heuristic policy HP, as defined by:

$$\text{gap}_{\text{HP}} = \frac{\text{LRAC}_{\text{HP}} - \text{LRAC}_{\text{OP}}}{\text{LRAC}_{\text{OP}}} \times 100\%,$$

where LRAC_{OP} denotes the long-run average cost incurred under the optimal policy and LRAC_{HP} denotes the long-run average cost incurred under the heuristic policy. The smaller the gap, the better the performance of the heuristic policy. Notice that the gap is scale-invariant, but not shift-invariant. That is to say, if we multiply the cost function c by a positive constant, the gap remains unchanged. But if we add a constant to the cost function c in our model, the gap changes. This is undesirable, especially when the cost consists of both the fixed cost and the variable cost, since this metric is not robust to changes in the fixed cost. In order to circumvent this problem, we introduce a metric that is both scale-invariant and shift-invariant.

Suppose the clinic currently uses the shortest-queue policy that results in LRAC_{SQ} and the cost of the optimal policy is LRAC_{OP}. Clearly $\text{LRAC}_{\text{OP}} \leq \text{LRAC}_{\text{SQ}}$. We propose a heuristic policy HP in place of SQ and the cost of HP is given by LRAC_{HP}. Then the relative efficiency η of HP over SQ is defined by:

$$\eta(\text{HP, SQ}) = \frac{\text{LRAC}_{\text{SQ}} - \text{LRAC}_{\text{HP}}}{\text{LRAC}_{\text{SQ}} - \text{LRAC}_{\text{OP}}} \times 100\%.$$

This metric tells us what fraction of the gap $\text{LRAC}_{\text{SQ}} - \text{LRAC}_{\text{OP}}$ is captured by the policy HP. The higher the relative efficiency, the better is the policy HP. The maximum relative efficiency is 100%, but it can be negative if HP is worse than SQ. It is clear that relative efficiency is both scale-invariant and shift-invariant.

In our numerical study, we vary p_1 and p_2 in parameter space $\mathcal{P}$ where $\mathcal{P} = \{(p_1, p_2)$: $p_1, p_2 \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}, p_1 + p_2 \leq 1\}$. Note that p_{12} can be computed by $p_{12} = 1 - p_1 - p_2$, but we do not show it explicitly in any table throughout this section. Table 2.2 displays the values of ρ_1 defined in Equation (2.9) for given p_1 and p_2, to be used in the calculations under the index policy and the randomized policy. We use the same parameters as in Section 2.4.1.

Table 2.3 exhibits how the four different policies perform as p_1 and p_2 vary. Each cell contains four numbers, where the first number shows the LRAC under the optimal policy (OP), the second indicates the difference between the LRAC under the index policy (IP) and OP, the third gives the difference between the LRAC under the shortest-queue policy (SQ) and IP, the fourth displays the difference between the LRAC under the randomized

Table 2.2: The values of ρ_1 given p_1 and p_2.

ρ_1	$p_1 = 0$	$p_1 = 0.2$	$p_1 = 0.4$	$p_1 = 0.6$	$p_1 = 0.8$	$p_1 = 1$
$p_2 = 0$	0.5	0.5	0.5	0.6	0.8	1
$p_2 = 0.2$	0.5	0.5	0.5	0.6	0.8	
$p_2 = 0.4$	0.5	0.5	0.5	0.6		
$p_2 = 0.6$	0.4	0.4	0.4			
$p_2 = 0.8$	0.2	0.2				
$p_2 = 1$	0					

policy (RP) and SQ. The zeros in anti-diagonal cells show that the LRAC under four different policies are the same. This is because there are no type-12 patients in these cases.

As one can see from Table 2.3, the ranking $\text{LRAC}_{\text{OP}} \leq \text{LRAC}_{\text{IP}} \leq \text{LRAC}_{\text{SQ}} \leq \text{LRAC}_{\text{RP}}$ is true under all sets of parameters and the index policy performs close to the optimal policy in terms of the long-run average cost. Theoretically, the index policy is known to perform better than the randomized policy given the fact that we apply the policy improvement algorithm on the randomized policy to come up with the index policy. But we cannot show that the index policy performs better than the shortest-queue policy analytically. The LRAC under the optimal policy increases with p_1 (p_2) for any fixed p_2 (p_1). Intuitively, the higher cost is due to the fact that the system becomes less flexible while the proportion of type-12 patients, $p_{12} = 1-p_1-p_2$ gets smaller. We also note that, for any fixed p_{12}, the system achieves lower cost when the number of type-1 arrivals is around the same as the number of type-2 arrivals. For example, the optimal LRAC in case $p_1 = 0.8, p_2 = 0, p_{12} = 0.2$ is 173.460, whereas the optimal LRAC in case $p_1 = 0.6, p_2 = 0.2, p_{12} = 0.2$ is 154.212. Apparently it is better for the system to see a more balanced arrivals between type-1 and type-2 patients.

One can also see that $\text{LRAC}_{\text{IP}} - \text{LRAC}_{\text{OP}}$ decreases with p_1 (p_2) for any fixed p_2 (p_1). A

Table 2.3: Long-run average cost under four different policies.

		$p_1 = 0$	$p_1 = 0.2$	$p_1 = 0.4$	$p_1 = 0.6$	$p_1 = 0.8$	$p_1 = 1$
	LRAC_{OP}						
	$\text{LRAC}_{\text{IP}} - \text{LRAC}_{\text{OP}}$						
	$\text{LRAC}_{\text{SQ}} - \text{LRAC}_{\text{IP}}$						
	$\text{LRAC}_{\text{RP}} - \text{LRAC}_{\text{SQ}}$						
$p_2 = 0$		125.012	130.535	139.110	152.203	173.460	208.751
		4.740	2.893	0.592	0.006	0.001	0
		12.696	12.418	10.196	4.586	0.627	0
		14.126	10.728	6.675	1.632	0.218	0
$p_2 = 0.2$		128.622	134.030	142.070	154.212	174.306	
		3.105	1.740	0.271	0.003	0	
		11.488	10.817	8.175	2.980	0	
		13.359	9.987	6.058	1.232	0	
$p_2 = 0.4$		133.644	139.141	146.910	158.427		
		1.707	0.828	0.086	0		
		9.995	8.690	5.281	0		
		11.228	7.914	4.297	0		
$p_2 = 0.6$		143.158	149.583	158.424			
		0.441	0.105	0			
		9.068	6.042	0			
		5.757	2.694	0			
$p_2 = 0.8$		165.539	174.296				
		0.046	0				
		6.667	0				
		2.043	0				
$p_2 = 1$		208.740					
		0					
		0					
		0					

reasonable explanation for this phenomenon is that a large part of the system is not under our control when the proportion of type 12 patients is low. This trend also holds for $\text{LRAC}_{\text{SQ}} - \text{LRAC}_{\text{IP}}$ and $\text{LRAC}_{\text{RP}} - \text{LRAC}_{\text{SQ}}$. Table 2.4 displays the relative efficiency of IP over SQ and

Table 2.4: Relative efficiencies assuming that on average 15 patients arrive per day.

η(IP,SQ) η(IP,RP)	$p_1 = 0$	$p_1 = 0.2$	$p_1 = 0.4$	$p_1 = 0.6$	$p_1 = 0.8$
$p_2 = 0$	72.81% 84.98%	81.10% 88.89%	94.51% 96.61%	99.88% 99.91%	99.89% 99.92%
$p_2 = 0.2$	78.72% 88.89%	86.15% 92.28%	96.79% 98.13%	99.89% 99.92%	
$p_2 = 0.4$	85.41% 92.56%	91.30% 95.25%	98.40% 99.11%		
$p_2 = 0.6$	95.36% 97.11%	98.28% 98.81%			
$p_2 = 0.8$	99.31% 99.47%				

the relative efficiency of IP over RP under different sets of parameters. Each cell contains two lines, the first being the relative efficiency of IP over SQ while the second being the relative efficiency of IP over RP. Note that when $p_1 + p_2 = 1$, no relative efficiency is defined because $p_{12} = 0$ leads to the situation that all four policies give rise to the same LRAC. We can see from Table 2.4, the relative efficiency of IP over SQ is over 72% in all cases. The relative efficiency of IP over RP is over 84% in all cases. In all cases we have the relationship eff(IP,SQ) $\leq$ eff(IP,RP). In Table 2.4, we also observe that both eff(IP,SQ) and eff(IP,RP) increase with p_1 (p_2) for any fixed p_2 (p_1). Under an environment where we have more type-12 patients, the scheduling is more likely out of our control and the relative efficiency of the index policy over another policy is lower.

2.7 Extension of the Base Model: Rejection is Allowed

In this section we study an extension of the base model, where the patient's request for an appointment can be rejected without incurring any cost except for the lost revenue.

2.7.1 Optimality Equation

Consider three cases of an arrival event when the system state is (i, j). If the arrival is of type-1, the decision is to give her an appointment on the next day or reject her, then the state changes to $(i+1, j)$ or stays in (i, j), respectively. If the arrival is of type-2, the decision is to give her an appointment on the day after next or reject her, then the state changes to $(i, j+1)$ or stays in (i, j), respectively. If the arrival is of type-12, she can be assigned to the next day with state changing to $(i+1, j)$, or the day after next with state changing to $(i, j+1)$, or she can be rejected with state staying in (i, j).

Using the dynamics above, the optimality equation can be written as:

$$h(i,j) + g = \alpha[p_{12} \min\{h(i+1,j), h(i,j+1), h(i,j)\} + p_1 \min\{h(i+1,j), h(i,j)\} \\ + p_2 \min\{h(i,j+1), h(i,j)\}] + (1-\alpha)[c(i) + h(j,0)]. \tag{2.12}$$

Suppose there is a solution (h, g) to Equation (2.12). Then g is the optimal average cost and the $h(\cdot,\cdot)$ is the bias.

The value iteration method to solve Equation (2.12) is given by:

$$\begin{aligned}v_k(i,j) = \alpha[&p_{12}\min\{v_{k-1}(i+1,j), v_{k-1}(i,j+1), v_{k-1}(i,j)\}\\&+ p_1\min\{v_{k-1}(i+1,j), v_{k-1}(i,j)\} + p_2\min\{v_{k-1}(i,j+1), v_{k-1}(i,j)\}]\\&+ (1-\alpha)[c(i) + v_{k-1}(j,0)],\end{aligned} \tag{2.13}$$

where we set $v_0(i,j) = 0$ for all $(i,j) \in S$.

The existence of a solution to Equation (2.12) and the convergence of the value iteration algorithm follows along similar lines as in Theorem 5. We write $d_t(i,j) = 1$ if the decision is to assign the patient to day $n+1$, $d_t(i,j) = 2$ if the decision is to assign the patient to day $n+2$, and $d_t(i,j) = 3$ if the decision is reject the patient. The optimal decisions are given as below.

- For a type-1 patient,

$$d_1(i,j) = \begin{cases} 1, \text{ if } h(i+1,j) \leq h(i,j); \\ 3, \text{ if } h(i+1,j) > h(i,j). \end{cases} \tag{2.14}$$

- For a type-2 patient,

$$d_2(i,j) = \begin{cases} 2, \text{ if } h(i,j+1) \leq h(i,j); \\ 3, \text{ if } h(i,j+1) > h(i,j). \end{cases} \tag{2.15}$$

- For a type-12 patient,

$$d_{12}(i,j) = \begin{cases} 1, \text{ if } h(i+1,j) \le \min\{h(i,j+1), h(i,j)\}; \\ 2, \text{ if } h(i,j+1) \le \min\{h(i+1,j), h(i,j)\}; \\ 3, \text{ if } h(i,j) \le \min\{h(i+1,j), h(i,j+1)\}. \end{cases} \tag{2.16}$$

2.7.2 Structural Properties

In this section, we study the structural properties of the optimal policy. We start with a fundamental assumption.

Assumption A. The function $c(\cdot)$ is convex and achieves its minimum at a positive integer m.

Using the parameter m we partition the state space $S = \{(i,j) : i \ge 0, j \ge 0\}$ into four regions: S_0, S_1, S_2, S_3, as shown in Figure 2.3. In mathematical terms, the four regions are defined as follows:

$$S_0 = \{(i,j)\colon 0 \le i \le m-1, 0 \le j \le m-1\};$$

$$S_1 = \{(i,j)\colon 0 \le i \le m-1, j \ge m\};$$

$$S_2 = \{(i,j)\colon i \ge m, 0 \le j \le m-1\};$$

$$S_3 = \{(i,j)\colon i \ge m, j \ge m\}.$$

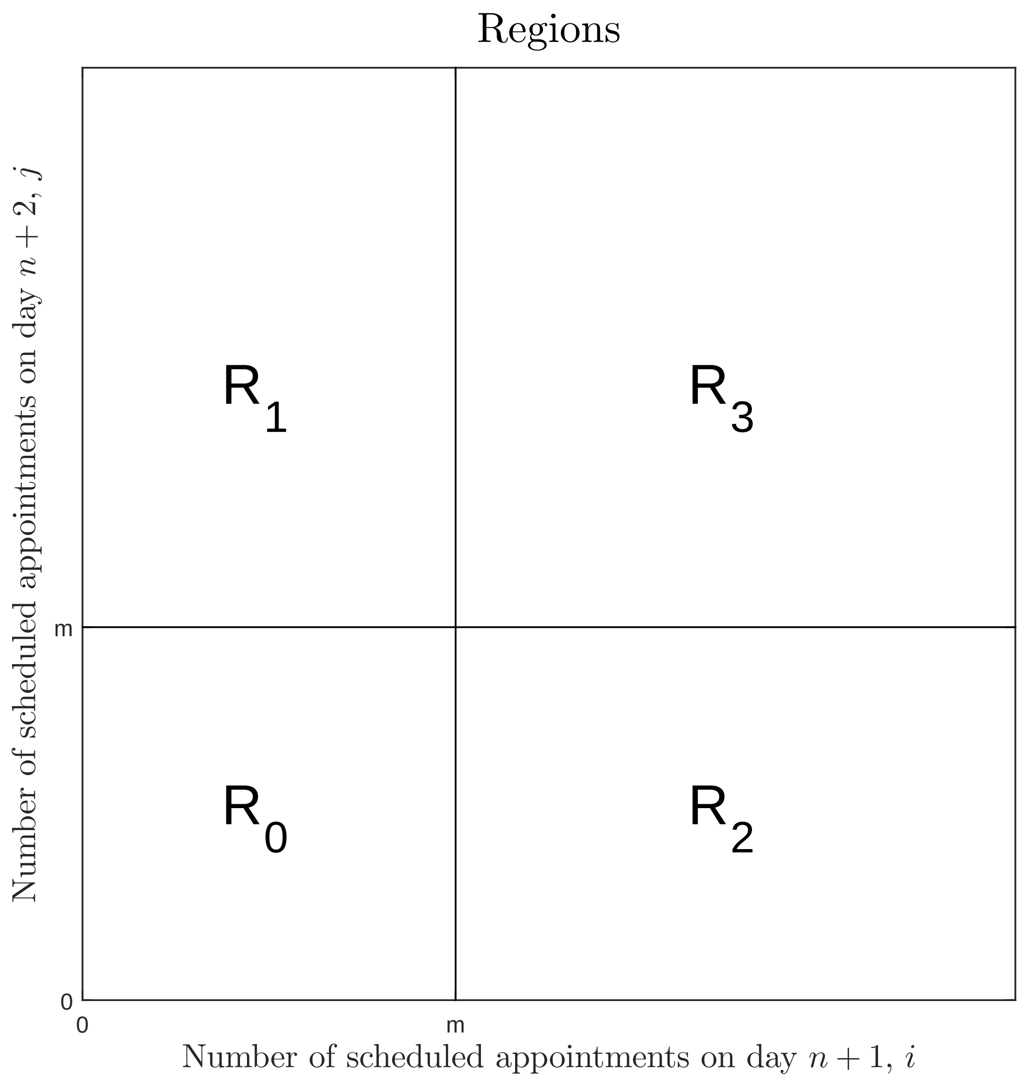

Figure 2.3: The regions classified based on the system states.

Next we study the properties of the bias $h(\cdot,\cdot)$. Consider a function $f : S \to \mathbb{R}$ with the properties

$$Inc(1) \text{ in } S_2 \cup S_3\text{:} \quad f(i+1,j) \geq f(i,j), \quad \forall (i,j) \in S_2 \cup S_3;$$

$$Dec(1) \text{ in } S_0 \cup S_1\text{:} \quad f(i+1,j) \leq f(i,j), \quad \forall (i,j) \in S_0 \cup S_1;$$

$$Inc(2) \text{ in } S_1 \cup S_3\text{:} \quad f(i,j+1) \geq f(i,j), \quad \forall (i,j) \in S_1 \cup S_3;$$

$$Dec(2) \text{ in } S_0 \cup S_2\text{:} \quad f(i,j+1) \leq f(i,j), \quad \forall (i,j) \in S_0 \cup S_2;$$

and

$$Convexity \text{ in } S_0\text{:} \begin{cases} f(i,j) + f(i+2,j) \geq 2f(i+1,j), & \forall (i+1,j) \in S_0; \\ f(i,j) + f(i,j+2) \geq 2f(i,j+1), & \forall (i,j+1) \in S_0; \end{cases}$$

$$Super \text{ in } S_0\text{: } f(i,j) + f(i+1,j+1) \geq f(i+1,j) + f(i,j+1), \quad \forall (i,j) \in S_0;$$

and $SuperC$ in S_0:

$$\begin{cases} f(i+2,j) + f(i,j+1) \geq f(i+1,j) + f(i+1,j+1), & \forall (i+1,j) \in S_0; \\ f(i+1,j) + f(i,j+2) \geq f(i,j+1) + f(i+1,j+1), & \forall (i,j+1) \in S_0. \end{cases}$$

We now redefine some operators as below.

$$T_{12}f(i,j) = \min\{f(i+1,j), f(i,j+1), f(i,j)\},$$

$$T_1 f(i,j) = \min\{f(i+1,j), f(i,j)\},$$

$$T_2 f(i,j) = \min\{f(i,j+1), f(i,j)\}.$$

Then v_k in Equation (2.13) can be rewritten as:

$$v_{k+1}(i,j) = T_\alpha(T_p(T_{12}v_k, T_1 v_k, T_2 v_k), T_c v_k)(i,j),$$

with $v_0(i,j) = 0$ for all $(i,j) \in S$. The operators T_p, T_α, T_c are defined in Section 2.4.

The following lemmas show the relevant properties of the operators.

Lemma 3. *The operator T_{12} satisfies:*

(a) $Inc(1)$ in $S_2 \cup S_3 \to Inc(1)$ in $S_2 \cup S_3$

(b) $Inc(1)$ in $S_2 \cup S_3, Dec(1)$ in $S_0 \cup S_1, Inc(2)$ in $S_1 \cup S_3, Dec(2)$ in $S_0 \cup S_2 \to Dec(1)$ in $S_0 \cup S_1$

(c) $Inc(2)$ in $S_1 \cup S_3 \to Inc(2)$ in $S_1 \cup S_3$

(d) $Inc(1)$ in $S_2 \cup S_3, Dec(1)$ in $S_0 \cup S_1, Inc(2)$ in $S_1 \cup S_3, Dec(2)$ in $S_0 \cup S_2 \to Dec(2)$ in $S_0 \cup S_2$

(e) $Inc(1)$ in $S_2 \cup S_3, Dec(1)$ in $S_0 \cup S_1, Inc(2)$ in $S_1 \cup S_3, Dec(2)$ in $S_0 \cup S_2$, $Convexity$, $Super$, $SuperC$ in $S_0 \to Convexity$ in S_0

(f) $Inc(1)$ in $S_2 \cup S_3, Dec(1)$ in $S_0 \cup S_1, Inc(2)$ in $S_1 \cup S_3, Dec(2)$ in $S_0 \cup S_2$, $Convexity$,

$Super$, $SuperC$ *in* $S_0 \rightarrow Super$ *in* S_0

(g) $Inc(1)$ *in* $S_2 \cup S_3, Dec(1)$ *in* $S_0 \cup S_1, Inc(2)$ *in* $S_1 \cup S_3, Dec(2)$ *in* $S_0 \cup S_2$, $Convexity$, $Super$, $SuperC$ *in* $S_0 \rightarrow SuperC$ *in* S_0.

Proof. (a). The result follows since the monotonicity is closed under the minimum operator.

(b). We need to show $T_{12}f(i+1,j) \leq T_{12}f(i,j)$ for $0 \leq i \leq m-1, j \geq 0$. We consider three cases:

Case (b)(1): $0 \leq i \leq m-2, j \geq 0$. The result follows since the monotonicity is closed under the minimum operator.

Case (b)(2): $i = m-1, 0 \leq j \leq m-1$. We need to show $T_{12}f(m,j) \leq T_{12}f(m-1,j)$. We have

$$Dec(2) \text{ in } S_0 \cup S_2 \Rightarrow f(m,j+1) \leq f(m,j);$$

$$Inc(1) \text{ in } S_2 \cup S_3 \Rightarrow f(m,j) \leq f(m+1,j).$$

Hence

$$T_{12}f(m,j) = \min\{f(m+1,j), f(m,j+1), f(m,j)\} = f(m,j+1).$$

Similary, we have

$$Dec(1) \text{ in } S_0 \cup S_1 \Rightarrow f(m,j) \leq f(m-1,j);$$

$$Dec(2) \text{ in } S_0 \cup S_2 \Rightarrow f(m-1,j+1) \leq f(m-1,j).$$

Hence

$$T_{12}f(m-1,j) = \min\{f(m,j), f(m-1,j+1), f(m-1,j)\}$$
$$= \min\{f(m,j), f(m-1,j+1)\}.$$

We know $f(m,j+1) \leq f(m,j)$ by $Dec(2)$ in $S_0 \cup S_2$ and $f(m,j+1) \leq f(m-1,j+1)$ by $Dec(1)$ in $S_0 \cup S_1$. Therefore

$$T_{12}f(m,j) = f(m,j+1) \leq \min\{f(m,j), f(m-1,j+1)\} = T_{12}f(m-1,j).$$

Case (b)(3): $i = m-1, j \geq m$. We need to show $T_{12}f(m,j) \leq T_{12}f(m-1,j)$. We have

$$Inc(1) \text{ in } S_2 \cup S_3 \Rightarrow f(m,j) \leq f(m+1,j);$$
$$Inc(2) \text{ in } S_1 \cup S_3 \Rightarrow f(m,j) \leq f(m,j+1).$$

Hence

$$T_{12}f(m,j) = \min\{f(m+1,j), f(m,j+1), f(m,j)\} = f(m,j).$$

We have

$$Dec(1) \text{ in } S_0 \cup S_1 \Rightarrow f(m,j) \leq f(m-1,j);$$
$$Inc(2) \text{ in } S_1 \cup S_3 \Rightarrow f(m-1,j) \leq f(m-1,j+1).$$

Therefore

$$T_{12}f(m-1,j) = \min\{f(m,j), f(m-1,j+1), f(m-1,j)\} = f(m,j).$$

The result follows.

(c). The result follows since the monotonicity is closed under the minimum operator.

(d). We need to show $T_{12}f(i,j+1) \leq T_{12}f(i,j)$ for $i \geq 0, 0 \leq j \leq m-1$. We consider three cases:

Case (d)(1): $i \geq 0, 0 \leq j \leq m-2$. The result follows since the monotonicity is closed under the minimum operator.

Case (d)(2): $0 \leq i \leq m-1, j = m-1$. We need to show $T_{12}f(i,m) \leq T_{12}f(i,m-1)$. We have

$$Dec(1) \text{ in } S_0 \cup S_1 \Rightarrow f(i+1,m) \leq f(i,m);$$

$$Inc(2) \text{ in } S_1 \cup S_3 \Rightarrow f(i,m) \leq f(i,m+1).$$

Hence

$$T_{12}f(i,m) = \min\{f(i+1,m), f(i,m+1), f(i,m)\} = f(i+1,m).$$

Similarly, we have

$$Dec(1) \text{ in } S_0 \cup S_1 \Rightarrow f(i+1, m-1) \leq f(i, m-1);$$

$$Dec(2) \text{ in } S_0 \cup S_2 \Rightarrow f(i, m) \leq f(i, m-1).$$

Therefore

$$\begin{aligned} T_{12}f(i, m-1) &= \min\{f(i+1, m-1), f(i, m), f(i, m-1)\} \\ &= \min\{f(i+1, m-1), f(i, m)\} \end{aligned}$$

We know $f(i+1, m) \leq f(i+1, m-1)$ by $Dec(2)$ in $S_0 \cup S_2$ and $f(i+1, m) \leq f(i, m)$ by $Dec(1)$ in $S_0 \cup S_1$.

$$T_{12}f(i, m) = f(i+1, m) \leq \min\{f(i+1, m-1), f(i, m)\} = T_{12}f(i, m-1).$$

Case (d)(3): $i \geq m, j = m-1$. We need to show $T_{12}f(i, m) \leq T_{12}f(i, m-1)$. We have

$$Inc(1) \text{ in } S_2 \cup S_3 \Rightarrow f(i, m) \leq f(i+1, m);$$

$$Inc(2) \text{ in } S_1 \cup S_3 \Rightarrow f(i, m) \leq f(i, m+1).$$

Hence

$$T_{12}f(i, m) = \min\{f(i+1, m), f(i, m+1), f(i, m)\} = f(i, m).$$

Similarly, we have

$$Dec(2) \text{ in } S_0 \cup S_2 \Rightarrow f(i,m) \leq f(i,m-1);$$

$$Inc(1) \text{ in } S_2 \cup S_3 \Rightarrow f(i,m-1) \leq f(i+1,m-1).$$

Therefore

$$T_{12}f(i,m-1) = \min\{f(i+1,m-1), f(i,m), f(i,m-1)\} = f(i,m).$$

The result follows.

(e). We show that $T_{12}f(i,j) + T_{12}f(i+2,j) \geq 2T_{12}f(i+1,j)$ for $0 \leq i \leq m-2, 0 \leq j \leq m-1$ and skip the proof that $T_{12}f(i,j) + T_{12}f(i,j+2) \geq 2T_{12}f(i,j+1)$ for $0 \leq i \leq m-1, 0 \leq j \leq m-2$ since it is similar. We consider four cases:

Case (e)(1) $i \leq m-3, j \leq m-1$. We know

$$T_{12}f(i,j) = \min\{f(i+1,j), f(i,j+1), f(i,j)\} = \min\{f(i+1,j), f(i,j+1)\}.$$

The T_{12} here coincides with the $T_{R(I)}$ in (Koole, 1998). The result follows.

Case (e)(2) $i \leq m-3, j = m$. We show

$$T_{12}f(i,m) + T_{12}f(i+2,m) \geq 2T_{12}f(i+1,m),$$

where we know

$$T_{12}f(i,m) = f(i+1,m);$$

$$T_{12}f(i+2,m) = f(i+3,m);$$

$$T_{12}f(i+1,m) = f(i+2,m).$$

The result follows from the $Convexity$ of f.

Case (e)(3) $i = m-2, j \leq m-1$. We show

$$T_{12}f(m-2,j) + T_{12}f(m,j) \geq 2T_{12}f(m-1,j),$$

where we know

$$T_{12}f(m-2,j) = \min\{f(m-1,j), f(m-2,j+1)\};$$

$$T_{12}f(m-1,j) = \min\{f(m,j), f(m-1,j+1)\};$$

$$T_{12}f(m,j) = f(m,j+1).$$

We consider two cases:

Case (e)(3)(i): $f(m-1,j) \geq f(m-2,j+1)$. Then $T_{12}f(m-2,j) = f(m-2,j+1)$. By the property $SuperC$ of f, we know

$$f(m,j) - f(m-1,j+1) \geq f(m-1,j) - f(m-2,j+1) \geq 0.$$

Hence $T_{12}f(m-1,j) = f(m-1,j+1)$. By the $Convexity$ of f, we have

$$\begin{aligned} T_{12}f(m-2,j) + T_{12}f(m,j) =& f(m-2,j+1) + f(m,j+1) \\ \geq& 2f(m-1,j+1) = 2T_{12}f(m-1,j). \end{aligned}$$

Case (e)(3)(ii): $f(m-1,j) < f(m-2,j+1)$. Then $T_{12}f(m-2,j) = f(m-1,j)$. By the $Super$ of f, we have

$$\begin{aligned} T_{12}f(m-2,j) + T_{12}f(m,j) &= f(m-1,j) + f(m,j+1) \\ &\geq f(m,j) + f(m-1,j+1) \\ &\geq 2\min\{f(m,j), f(m-1,j+1)\} \\ &= 2T_{12}f(m-1,j). \end{aligned}$$

Case (e)(4) $i = m-2, j = m$. We show

$$T_{12}f(m-2,m) + T_{12}f(m,m) \geq 2T_{12}f(m-1,m),$$

where we know

$$\begin{aligned} T_{12}f(m-2,m) &= f(m-1,m); \\ T_{12}f(m-1,m) &= f(m,m); \\ T_{12}f(m,m) &= f(m,m). \end{aligned}$$

The result follows from $f(m-1,m) \geq f(m,m)$ by $Dec(1)$ in $S_0 \cup S_1$.

(f). We show that $T_{12}f(i,j) + T_{12}f(i+1,j+1) \geq T_{12}f(i+1,j) + T_{12}f(i,j+1)$ for $0 \leq i,j \leq m-1$. We consider four cases:

Case (f)(1) $i \leq m-2, j \leq m-2$. We know $T_{12}f(i,j) = \min\{f(i+1,j), f(i,j+1)\}$ coincides with the $T_{R(I)}$ in (Koole, 1998). The result follows.

Case (f)(2) $i = m-1, j \leq m-2$. We show

$$T_{12}f(m-1,j) + T_{12}f(m,j+1) \geq T_{12}f(m,j) + T_{12}f(m-1,j+1),$$

where we know

$$\begin{aligned}
T_{12}f(m-1,j) &= \min\{f(m,j), f(m-1,j+1)\}; \\
T_{12}f(m,j+1) &= f(m,j+2); \\
T_{12}f(m,j) &= f(m,j+1); \\
T_{12}f(m-1,j+1) &= \min\{f(m,j+1), f(m-1,j+2)\}.
\end{aligned}$$

We consider two cases:

Case (f)(2)(i): $f(m,j) \geq f(m-1,j+1)$. Then $T_{12}f(m-1,j) = f(m-1,j+1)$. By the

$Super$ of f, we know

$$\begin{aligned}T_{12}f(m-1,j)+T_{12}f(m,j+1) &= f(m-1,j+1)+f(m,j+2)\\ &\geq f(m,j+1)+f(m-1,j+2) \text{ (by } Super)\\ &\geq f(m,j+1)+\min\{f(m,j+1),f(m-1,j+2)\}\\ &= T_{12}f(m,j)+T_{12}f(m-1,j+1).\end{aligned}$$

Case (f)(2)(ii): $f(m,j) < f(m-1,j+1)$. Then $T_{12}f(m-1,j) = f(m,j)$. By the $SuperC$ of f, we know

$$f(m,j+1)-f(m-1,j+2) \leq f(m,j)-f(m-1,j+1) < 0.$$

Hence $T_{12}f(m-1,j+1) = f(m,j+1)$. By the $Convexity$ of f, we have

$$\begin{aligned}T_{12}f(m-1,j)+T_{12}f(m,j+1) =&f(m,j)+f(m,j+2)\\ \geq&f(m,j+1)+f(m,j+1)\\ =&T_{12}f(m,j)+T_{12}f(m-1,j+1).\end{aligned}$$

Case (f)(3) $i \leq m-2, j = m-1$. We show

$$T_{12}f(i,m-1)+T_{12}f(i+1,m) \geq T_{12}f(i+1,m-1)+T_{12}f(i,m),$$

where we know

$$T_{12}f(i,m-1) = \min\{f(i+1,m-1), f(i,m)\};$$

$$T_{12}f(i+1,m) = f(i+2,m);$$

$$T_{12}f(i+1,m-1) = \min\{f(i+2,m-1), f(i+1,m)\};$$

$$T_{12}f(i,m) = f(i+1,m).$$

We know

$$\begin{aligned}
&T_{12}f(i,m-1) + T_{12}f(i+1,m)\\
=&\min\{f(i+1,m-1), f(i,m)\} + f(i+2,m)\\
=&\min\{f(i+1,m-1) + f(i+2,m), f(i,m) + f(i+2,m)\}\\
\geq&\min\{f(i+1,m-1) + f(i+2,m), 2f(i+1,m)\} \text{ (by } Convexity)\\
\geq&\min\{f(i+2,m-1) + f(i+1,m), 2f(i+1,m)\} \text{ (by } Super)\\
=&\min\{f(i+2,m-1), f(i+1,m)\} + f(i+1,m)\\
=&T_{12}f(i+1,m-1) + T_{12}f(i,m).
\end{aligned}$$

Case (f)(4) $i = m-1, j = m-1$. We show

$$T_{12}f(m-1,m-1) + T_{12}f(m,m) \geq T_{12}f(m,m-1) + T_{12}f(m-1,m),$$

where we know

$$T_{12}f(m-1,m-1) = \min\{f(m,m-1), f(m-1,m)\};$$
$$T_{12}f(m,m) = f(m,m);$$
$$T_{12}f(m,m-1) = f(m,m);$$
$$T_{12}f(m-1,m) = f(m,m).$$

We know

$$\begin{aligned}T_{12}f(m-1,m-1) + T_{12}f(m,m) &= \min\{f(m,m-1), f(m-1,m)\} + f(m,m)\\ &\geq f(m,m) + f(m,m) \text{ (by } Dec(1) \text{ and } Dec(2))\\ &= T_{12}f(m,m-1) + T_{12}f(m-1,m).\end{aligned}$$

(g). We show that $T_{12}f(i+2,j) + T_{12}f(i,j+1) \geq T_{12}f(i+1,j) + T_{12}f(i+1,j+1)$ for $0 \leq i \leq m-2, 0 \leq j \leq m-1$. We skip the proof of $T_{12}f(i+1,j) + T_{12}f(i,j+2) \geq T_{12}f(i,j+1) + T_{12}f(i+1,j+1)$ for $0 \leq i \leq m-1, 0 \leq j \leq m-2$. We consider four cases:

Case (g)(1) $i \leq m-3, j \leq m-2$. We know $T_{12}f(i,j) = \min\{f(i+1,j), f(i,j+1)\}$ coincides with the $T_{R(I)}$ in (Koole, 1998). The result follows.

Case (g)(2) $i = m-2, j \leq m-2$. We show

$$T_{12}f(m,j) + T_{12}f(m-2,j+1) \geq T_{12}f(m-1,j) + T_{12}f(m-1,j+1),$$

where we know

$$T_{12}f(m,j) = f(m,j+1);$$

$$T_{12}f(m-2,j+1) = \min\{f(m-1,j+1), f(m-2,j+2)\};$$

$$T_{12}f(m-1,j) = \min\{f(m,j), f(m-1,j+1)\};$$

$$T_{12}f(m-1,j+1) = \min\{f(m,j+1), f(m-1,j+2)\}.$$

We consider two cases:

Case (g)(2)(i): $f(m-1,j+1) < f(m-2,j+2)$. Then $T_{12}f(m-2,j+1) = f(m-1,j+1)$.

We have

$$\begin{aligned}
&T_{12}f(m,j) + T_{12}f(m-2,j+1) \\
=&f(m,j+1) + f(m-1,j+1) \\
\geq&\min\{f(m,j+1), f(m-1,j+2)\} + \min\{f(m,j), f(m-1,j+1)\} \\
=&T_{12}f(m-1,j+1) + T_{12}f(m-1,j).
\end{aligned}$$

Case (g)(2)(ii): $f(m-1,j+1) \geq f(m-2,j+2)$. Then $T_{12}f(m-2,j+1) = f(m-2,j+2)$.

We have

$$\begin{aligned}
SuperC &\Rightarrow f(m,j+1) - f(m-1,j+2) \geq f(m-1,j+1) - f(m-2,j+2) \geq 0 \\
&\Rightarrow T_{12}f(m-1,j+1) = f(m-1,j+2),
\end{aligned}$$

and

$$SuperC \Rightarrow f(m,j) - f(m-1,j+1) \geq f(m,j+1) - f(m-1,j+2) \geq 0$$

$$\Rightarrow T_{12}f(m-1,j) = f(m-1,j+1).$$

We have

$$\begin{aligned}
&T_{12}f(m,j) + T_{12}f(m-2,j+1)\\
=&f(m,j+1) + f(m-2,j+2)\\
\geq&f(m-1,j+2) + f(m-1,j+1) \text{ (by } SuperC)\\
=&T_{12}f(m-1,j+1) + T_{12}f(m-1,j).
\end{aligned}$$

Case (g)(3) $i \leq m-3, j = m-1$. We show

$$T_{12}f(i+2,m-1) + T_{12}f(i,m) \geq T_{12}f(i+1,m-1) + T_{12}f(i+1,m),$$

where we know

$$T_{12}f(i+2,m-1) = \min\{f(i+3,m-1), f(i+2,m)\};$$

$$T_{12}f(i,m) = f(i+1,m);$$

$$T_{12}f(i+1,m-1) = \min\{f(i+2,m-1), f(i+1,m)\};$$

$$T_{12}f(i+1,m) = f(i+2,m).$$

We consider two cases:

Case (g)(3)(i): $f(i+3,m-1) < f(i+2,m)$. Then $T_{12}f(i+2,m-1) = f(i+3,m-1)$. We have

$$\begin{aligned}
SuperC &\Rightarrow f(i+2,m-1) - f(i+1,m) \le f(i+3,m-1) - f(i+2,m) < 0 \\
&\Rightarrow T_{12}f(i+1,m-1) = f(i+2,m-1).
\end{aligned}$$

Hence

$$\begin{aligned}
&T_{12}f(i+2,m-1) + T_{12}f(i,m) \\
=&f(i+3,m-1) + f(i+1,m) \\
\ge&f(i+2,m-1) + f(i+2,m) \text{ (by } SuperC) \\
=&T_{12}f(i+1,m-1) + T_{12}f(i+1,m).
\end{aligned}$$

Case (g)(3)(ii): $f(i+3,m-1) \ge f(i+2,m)$. Then $T_{12}f(i+2,m-1) = f(i+2,m)$. We have

$$\begin{aligned}
T_{12}f(i+2,m-1) + T_{12}f(i,m) &= f(i+2,m) + f(i+1,m) \\
&\ge f(i+2,m) + \min\{f(i+2,m-1), f(i+1,m)\} \\
&= T_{12}f(i+1,m) + T_{12}f(i+1,m-1).
\end{aligned}$$

Case (g)(4) $i = m-2, j = m-1$. We show

$$T_{12}f(m,m-1) + T_{12}f(m-2,m) \geq T_{12}f(m-1,m-1) + T_{12}f(m-1,m),$$

where we know

$$\begin{aligned} T_{12}f(m,m-1) &= f(m,m); \\ T_{12}f(m-2,m) &= f(m-1,m); \\ T_{12}f(m-1,m-1) &= \min\{f(m,m-1), f(m-1,m)\}; \\ T_{12}f(m-1,m) &= f(m,m). \end{aligned}$$

Hence

$$\begin{aligned} & T_{12}f(m,m-1) + T_{12}f(m-2,m) \\ =& f(m,m) + f(m-1,m) \\ \geq& f(m,m) + \min\{f(m,m-1), f(m-1,m)\} \\ =& T_{12}f(m-1,m) + T_{12}f(m-1,m-1). \end{aligned}$$

□

Lemma 4. *The operators T_1, T_2 satisfy (a) to (g) in Lemma 3.*

Proof. Due to the symmetry, we show the proof for operator T_1 and skip the proof for operator T_2. Given the properties f has, we know

$$T_1 f(i,j) = \min\{f(i+1,j), f(i,j)\} = \begin{cases} f(i+1,j), & \forall (i,j) \in S_0 \cup S_1, \\ f(i,j), & \forall (i,j) \in S_2 \cup S_3. \end{cases}$$

We prove (f) and skip the rest due to the similarity. We show that $T_1 f(i,j) + T_1 f(i+1,j+1) \geq T_1 f(i+1,j) + T_1 f(i,j+1)$ for any $0 \leq i,j \leq m-1$. We consider four cases:

Case (f)(1): $0 \leq i \leq m-2, 0 \leq j \leq m-2$. We have

$$\begin{aligned} T_1 f(i,j) + T_1 f(i+1,j+1) &= f(i+1,j) + f(i+2,j+1) \\ &\geq f(i+2,j) + f(i+1,j+1) \text{ (by } Super) \\ &= T_1 f(i+1,j) + T_1 f(i,j+1). \end{aligned}$$

Case (f)(2): $i = m-1, 0 \leq j \leq m-2$. We need to show $T_1 f(m-1,j) + T_1 f(m,j+1) \geq T_1 f(m,j) + T_1 f(m-1,j+1)$. We have

$$\begin{aligned} T_1 f(m-1,j) + T_1 f(m,j+1) &= f(m,j) + f(m,j+1) \\ &= T_1 f(m,j) + T_1 f(m-1,j+1) \end{aligned}$$

Case (f)(3): $0 \leq i \leq m-2, j = m-1$. We need to show $T_1 f(i,m-1) + T_1 f(i+1,m) \geq$

$T_1 f(i+1, m-1) + T_1 f(i, m)$. We have

$$\begin{aligned} T_1 f(i, m-1) + T_1 f(i+1, m) &= f(i+1, m-1) + f(i+2, m) \\ &\geq f(i+2, m-1) + f(i+1, m) \text{ (by } Super) \\ &= T_1 f(i+1, m-1) + T_1 f(i, m). \end{aligned}$$

Case (f)(4): $i = m-1, j = m-1$. We need to show $T_1 f(m-1, m-1) + T_1 f(m, m) \geq T_1 f(m, m-1) + T_1 f(m-1, m)$. This follows because

$$\begin{aligned} &T_1 f(m-1, m-1) + T_1 f(m, m) \\ =&f(m, m-1) + f(m, m) \\ =&T_1 f(m, m-1) + T_1 f(m-1, m). \end{aligned}$$

□

Lemma 5. *The operators T_p, T_α satisfy:*

(a) $Inc(1)$ in $S_2 \cup S_3 \to Inc(1)$ in $S_2 \cup S_3$

(b) $Dec(1)$ in $S_0 \cup S_1 \to Dec(1)$ in $S_0 \cup S_1$

(c) $Inc(2)$ in $S_1 \cup S_3 \to Inc(2)$ in $S_1 \cup S_3$

(d) $Dec(2)$ in $S_0 \cup S_2 \to Dec(2)$ in $S_0 \cup S_2$

(e) $Convexity$ in $S_0 \to Convexity$ in S_0

(f) $Super$ in $S_0 \to Super$ in S_0

(g) $SuperC$ in $S_0 \to SuperC$ in S_0.

Proof. Given the definition of T_p and T_α, the conditions (a) to (g) follow, since the properties are closed under convex combination. $\square$

Lemma 6. *Under the Assumption A. The operator T_c satisfies:*

(a) $Inc(1)$ in $S_2 \cup S_3 \rightarrow Inc(1)$ in $S_2 \cup S_3$

(b) $Dec(1)$ in $S_0 \cup S_1 \rightarrow Dec(1)$ in $S_0 \cup S_1$

(c) $Inc(1)$ in $S_2 \cup S_3 \rightarrow Inc(2)$ in $S_1 \cup S_3$

(d) $Dec(1)$ in $S_0 \cup S_1 \rightarrow Dec(2)$ in $S_0 \cup S_2$

(e) $Convexity, Super, SuperC$ in $S_0 \rightarrow Convexity$ in S_0

(f) $Convexity, Super, SuperC$ in $S_0 \rightarrow Super$ in S_0

(g) $Convexity, Super, SuperC$ in $S_0 \rightarrow SuperC$ in S_0.

Proof. We know $T_c f(i,j) = c(i) + f(j,0)$ and $c(x)$ is a convex function which achieves its minimum at $x = m$, hence (a) and (b) follow. For (c), if f is $Inc(1)$ in $S_2 \cup S_3$, then $f(i+1,j) \geq f(i,j), \forall i \geq m, j \geq 0$. For any $j \geq m$, we have $T_c f(i,j+1) = c(i) + f(j+1,0) \geq c(i) + f(j,0) = T_c f(i,j)$. Hence $T_c f$ is $Inc(2)$ in $S_1 \cup S_3$. For (d), if f is $Dec(1)$ in $S_0 \cup S_1$, then $f(i+1,j) \leq f(i,j), \forall i \leq m-1, j \geq 0$. For any $j \leq m-1$, we have $T_c f(i,j+1) = c(i) + f(j+1,0) \leq c(i) + f(j,0) = T_c f(i,j)$. Hence $T_c f$ is $Dec(2)$ in $S_0 \cup S_2$. For (e), the convexity of $T_c f$ follows directly from the convexity of f. (f) is trivially shown given the definition of the operator T_c. For (g), we have

$$\begin{aligned} T_c f(i+2,j) + T_c f(i,j+1) &= c(i+2) + f(j,0) + c(i) + f(j+1,0) \\ &\geq c(i+1) + f(j,0) + c(i+1) + f(j+1,0) \\ &= T_c f(i+1,j) + T_c f(i+1,j+1), \end{aligned}$$

and

$$\begin{aligned} T_c f(i+1,j) + T_c f(i,j+2) &= c(i+1) + f(j,0) + c(i) + f(j+2,0) \\ &\geq c(i) + f(j+1,0) + c(i+1) + f(j+1,0) \\ &= T_c f(i,j+1) + T_c f(i+1,j+1). \end{aligned}$$

□

Next we prove the properties of the bias $h(\cdot,\cdot)$.

Theorem 10. *Under Assumption A, the bias $h(\cdot,\cdot)$ satisfies properties $Inc(1)$ in $S_2 \cup S_3$, $Dec(1)$ in $S_0 \cup S_1$, $Inc(2)$ in $S_1 \cup S_3$, $Dec(2)$ in $S_0 \cup S_2$ and $Convexity$, $Super$, $SuperC$ in S_0.*

Proof. By the lemmas above, the induction hypothesis gives rise to the fact that the functions v_k satisfy these properties. By Theorems 4 and 5, the bias $h(\cdot,\cdot)$ has these properties. □

A direct implication of Theorem 10 is given in Corollary 2, which gives the characteristics of the optimal decisions in the optimal policy.

Corollary 2. *The optimal decisions are given as follows.*

- *For a type-1 patient, we have*

$$d_1(i,j) = \begin{cases} 1, & \textit{if } i \leq m-1. \\ 3, & \textit{if } i \geq m. \end{cases}$$

- *For a type-2 patient, we have*

$$d_2(i,j) = \begin{cases} 2, & \textit{if } j \leq m-1. \\ 3, & \textit{if } j \geq m. \end{cases}$$

- *For a type-12 patient, we have*

$$d_{12}(i,j) = \begin{cases} d, & \textit{if } (i,j) \in S_d, \textit{ where } d = 1,2,3. \\ 1 \textit{ or } 2, & \textit{if } (i,j) \in S_0. \end{cases}$$

and for $(i,j) \in S_0$*,*

$$d_{12}(i,j) = 1 \Rightarrow d_{12}(i,j+1) = 1, \quad d_{12}(i,j) = 2 \Rightarrow d_{12}(i+1,j) = 2.$$

Proof. Follows directly from Theorem 10 and equations (2.14) to (2.16). □

Theorem 10 and Corollary 2 yield the following structure of the optimal policy.

Theorem 11. *Under Assumption A, for $0 \leq i \leq m-1$, there exists a critical number $j^*(i) \leq m$ and for $0 \leq j \leq m-1$ there exists a critical number $i^*(j) \leq m$ such that*

$$d_{12}(i,j) = \begin{cases} 1, & \text{if } i \leq m-1, j \geq j^*(i) \\ 2, & \text{if } i \geq i^*(j), j \leq m-1 \\ 3, & \text{if } i \geq m, j \geq m. \end{cases}$$

2.7.3 Numerical Illustration

We now conduct a numerical study to illustrate the structural properties of the optimal policy. We use the same parameters as in Section 2.4.1. The only difference is that rejection is available here. Figure 2.4 displays the optimal decision for a type-12 patient in each state. We do not display the optimal decisions for type-1 or type-2 patient because the policy is explicitly given by Corollary 2. Figure 2.4 indicates that it is optimal to stop scheduling a patient on any day once the number of scheduled appointments on that day reaches m.

2.7.4 Heuristic Policies

In this section we briefly discuss how the heuristic policies proposed in Section 2.5 can be applied to the extension of the base model. If the state (i,j) is in region $S_1 \cup S_2 \cup S_3$, we know the optimal decisions from Corollary 2. We implement these optimal decisions. If the state (i,j) is in region S_0, then we follow the heuristic policies proposed in Section 2.5, but only to do the calculations for states (i,j) in region S_0. The numerical performance of these

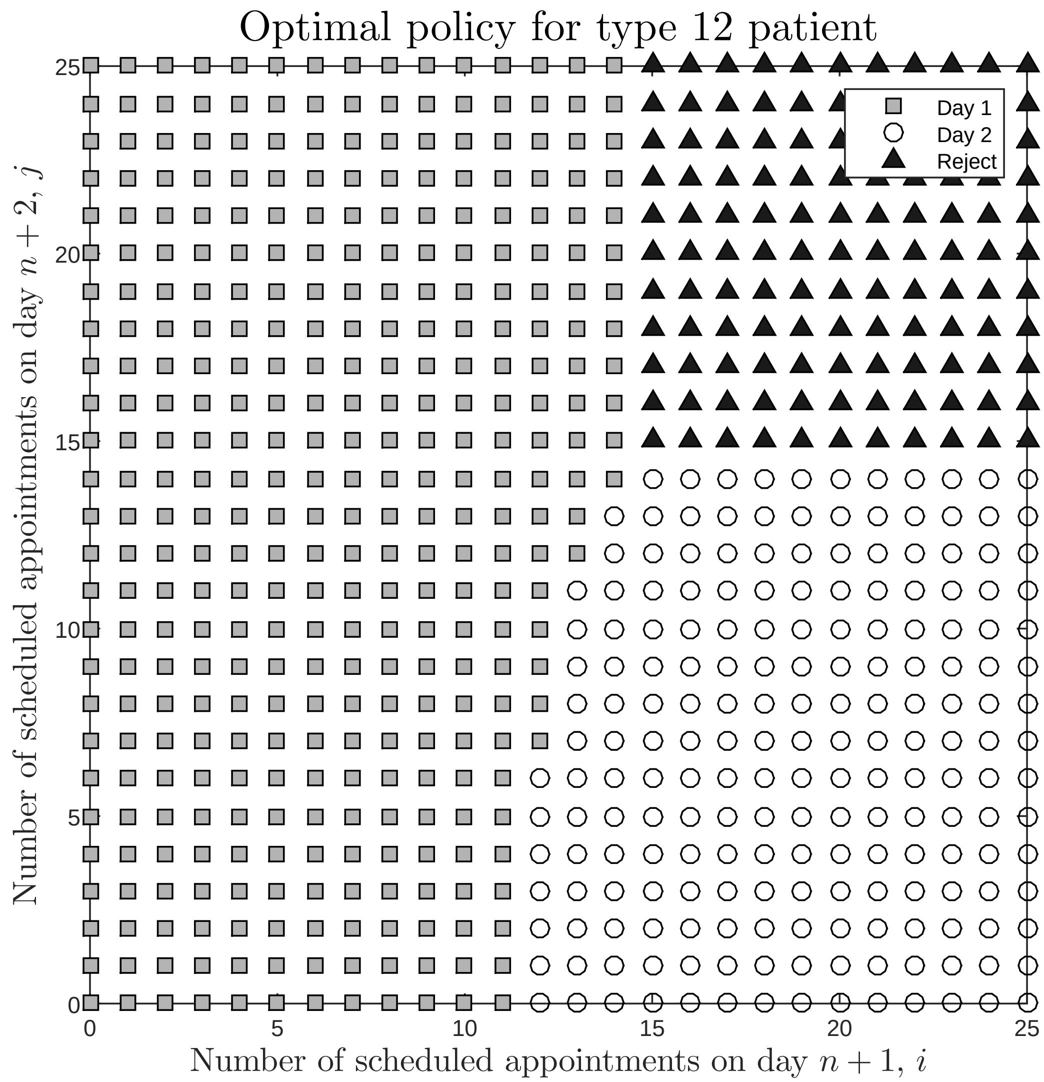

Figure 2.4: The optimal policy for type-12 patients when the rejection option is available.

policies is similar to the one reported in Section 2.6, so we chose to omit these results.

2.8 Extension of the Base Model: Multi-day Scheduling

We now generalize the base model to more-than-two-day scheduling horizon. As one can imagine, the number of types of patients increases exponentially with the length of scheduling horizon. It becomes much harder to compute the optimal policy in this generalized case due to the large state space. However, we expect the index policy to outperform the shortest-queue policy. We also relax the assumption that the number of arrivals in a day is Geometrically distributed. We consider both Poisson distribution and Negative Binomial distribution as the potential candidates for the arrival process.

In this section, we first present the general framework for appointment scheduling problem with the more-than-two-day horizon and Geometric arrivals. We use a simple case, three-day appointment scheduling horizon with three types of patients, to illustrate that the index policy outperforms the shortest-queue policy. We then generalize the arrival processes from Geometric distribution to Poisson distribution and Negative Binomial distribution. We show that our index policy is quite robust and performs well even if the Geometric arrival assumption is violated. We conclude this section by proposing two variants of the index policy for benchmarking.

2.8.1 General Framework and A Simple Case

Consider the appointment scheduling problem with T-day horizon. We have $K \leq 2^T$ types of patients who arrive on day n, where type-k patient prefers days in the set $\mathscr{A}_k$ for $k = 1, 2, ..., K$. We know $\mathscr{A}_k \subset \{1, 2, ..., T\}$ for $k = 1, 2, ..., K$. We assume that the number of arrivals on each day follows a Geometric distribution, same as in Section 2.3. With probability p_k, the arrival belongs to type-k for $k = 1, 2, ..., K$. We have $\sum_{k=1}^{K} p_k = 1$. Let $x = (x_1, ..., x_T)$ denote the system state just before an event on day n, where x_t denotes the number of appointments scheduled on day $n + t$ for $t = 1, ..., T$. If the event is an arrival (this occurs with probability α), the decision must be made such that the arrival is scheduled on the day $i \in \mathscr{A}_k$ and the system state transits to $x + e_i$. If the event is a change-of-day (this occurs with probability $1 - \alpha$), the system state will transit to $y = (x_2, ..., x_T, 0)$. By the standard theory of the dynamic programming, the optimality equation is given by:

$$h(x) + g = \alpha \sum_{k=1}^{K} p_k \min_{i \in \mathscr{A}_k} \{h(x + e_i)\} + (1 - \alpha)[c(x_1) + h(y)]$$

where $x = (x_1, x_2, ..., x_T)$ and $y = (x_2, ..., x_T, 0)$. Here g is the optimal long run average cost and $h(\cdot, \cdot)$ is the bias under the optimal policy.

To develop the index policy, we start from a randomized policy γ. The policy γ assigns type-k patient to day $n + j$ with probability θ_{kj} where $k = 1, 2, ..., K$ and $j \in \mathscr{A}_k$. We have $\sum_{j \in \mathscr{A}_k} \theta_{kj} = 1$ and $\theta_{kj} = 0$ for $j \notin \mathscr{A}_k$. We use A_n to denote the number of patients arriving on day n. Among those patients, we assign $A_{n,t}$ patients on day $n + t$. From the proof of Theorem 8, we know

$$A_{n,t} \sim \mathrm{G}\left(\frac{\alpha \rho_t}{1 - \alpha(1 - \rho_t)}\right), \quad t = 1, ..., T \tag{2.17}$$

where

$$\rho_t = \sum_{k=1}^{K} p_k \theta_{kt}.$$

A natural question is how to choose θ_{kj}. We will revisit this question later.

Suppose at any time on day n, the system state is $(x_1, ..., x_T)$. Then by the end of day n, we know the system state is

$$(x_1 + A_{n,1}, x_2 + A_{n,2}, x_3 + A_{n,3}, ..., x_T + A_{n,T}).$$

At the beginning of day $n+1$, we incur the cost of $c(x_1 + A_{n,1})$ and the system state becomes

$$(x_2 + A_{n,2}, x_3 + A_{n,3}, ..., x_T + A_{n,T}, 0).$$

By the end of day $n+1$, the system state is

$$(x_2 + A_{n,2} + A_{n+1,1}, x_3 + A_{n,3} + A_{n+1,2}, ..., x_T + A_{n,T} + A_{n+1,T-1}, A_{n+1,T}).$$

At the beginning of day $n+2$, we incur the cost of $c(x_2 + A_{n,2} + A_{n+1,1})$ and the system state becomes

$$(x_3 + A_{n,3} + A_{n+1,2}, ..., x_T + A_{n,T} + A_{n+1,T-1}, A_{n+1,T}, 0).$$

Similarly, at the beginning of day $n+T$, we incur the cost of $c(x_T + A_{n,T} + A_{n+1,T-1} + \cdots + A_{n+T-1,1})$ and the system state becomes

$$(A_{n+1,T} + \cdots + A_{n+T-1,2}, ..., A_{n+T-1,T}, 0).$$

Therefore we can write

$$\begin{aligned} h^{\gamma}(x) =& \mathrm{E}[c(x_1 + A_{n,1})] + \mathrm{E}[c(x_2 + A_{n,2} + A_{n+1,1})] + \cdots \\ &+ \mathrm{E}[c(x_T + A_{n,T} + A_{n+1,T-1} + \cdots + A_{n+T-1,1})] \\ &+ h^{\gamma}(A_{n+1,T} + \cdots + A_{n+T-1,2}, ..., A_{n+T-1,T}, 0). \end{aligned}$$

Hence the index for day k is

$$\begin{aligned} I_k(x_k) =& \mathrm{E}[c(x_k + 1 + A_{n,k} + A_{n+1,k-1} + \cdots + A_{n+k-1,1})] \\ &- \mathrm{E}[c(x_k + A_{n,k} + A_{n+1,k-1} + \cdots + A_{n+k-1,1})] \end{aligned}$$

Since $A_{n,k}$ has same distribution for all n, so we write

$$B_k = A_{n,k} + A_{n,k-1} + \cdots + A_{n,1}.$$

Thus

$$I_k(x_k) = \mathrm{E}[c(x_k + 1 + B_k)] - \mathrm{E}[c(x_k + B_k)].$$

This general framework works for all T and K. Next we use an example to study the performance of the index policy numerically.

We consider a three-day appointment scheduling problem. There are three types of patients: a type-1 patient requests an appointment on day 1, 2, or 3; a type-2 patient requests an appointment on day 2 or 3; a type-3 patient requests an appointment on day 3. That is to say, $\mathscr{A}_1 = \{1, 2, 3\}, \mathscr{A}_2 = \{2, 3\}, \mathscr{A}_3 = \{3\}$. In this setting, the index function under the general framework reduces to

$$I_k(x_k) = \mathrm{E}[c(x_k + 1 + B_k)] - \mathrm{E}[c(x_k + B_k)], \quad k = 1, 2, 3$$

where $B_1 = A_{n,1}, B_2 = A_{n,1}+A_{n,2}, B_3 = A_{n,1}+A_{n,2}+A_{n,3} = A_n$. We know the distribution $A_{n,t}$ from Equation (2.17). To implement the index policy, we need to know the values of ρ_1, ρ_2, ρ_3 to compute the distribution of B_k's. We use the same method to find these values as in Section 2.5.2. That is, ρ_1, ρ_2, ρ_3 are the solutions to the optimization problem

$$\begin{aligned}
\min\ & (\rho_1 - \frac{1}{3})^2 + (\rho_2 - \frac{1}{3})^2 + (\rho_3 - \frac{1}{3})^2 \\
\text{s.t.}\ & 0 \leq \rho_1 \leq p_1, \\
& 0 \leq \rho_2 \leq p_1 + p_2, \\
& p_3 \leq \rho_3 \leq 1, \\
& \rho_1 + \rho_2 + \rho_3 = 1.
\end{aligned}$$

We solve this quadratic program to obtain the values of ρ_1, ρ_2, ρ_3 given the inputs p_1, p_2, p_3. Table 2.5 includes that the values of ρ_1, ρ_2 under different p_1, p_2. Note that $p_3 = 1 - p_1 - p_2$ and $\rho_3 = 1 - \rho_1 - \rho_2$. Hence the values of p_3 and ρ_3 are known given the values of p_1, p_2 and ρ_1, ρ_2.

We now propose a measure to quantify the performance of the index policy compared to

(ρ_1, ρ_2)	$p_1 = 0$	$p_1 = 0.2$	$p_1 = 0.4$	$p_1 = 0.6$	$p_1 = 0.8$	$p_1 = 1$
$p_2 = 0$	$(0, 0)$	$(0.1, 0.1)$	$(0.2, 0.2)$	$(0.3, 0.3)$	$(\frac{1}{3}, \frac{1}{3})$	$(\frac{1}{3}, \frac{1}{3})$
$p_2 = 0.2$	$(0, 0.2)$	$(0.2, 0.2)$	$(0.3, 0.3)$	$(\frac{1}{3}, \frac{1}{3})$	$(\frac{1}{3}, \frac{1}{3})$	
$p_2 = 0.4$	$(0, 0.4)$	$(0.2, 0.4)$	$(\frac{1}{3}, \frac{1}{3})$	$(\frac{1}{3}, \frac{1}{3})$		
$p_2 = 0.6$	$(0, 0.5)$	$(0.2, 0.4)$	$(\frac{1}{3}, \frac{1}{3})$			
$p_2 = 0.8$	$(0, 0.5)$	$(0.2, 0.4)$				
$p_2 = 1$	$(0, 0.5)$					

Table 2.5: The values of ρ_1, ρ_2 under different p_1, p_2.

the shortest-queue policy. Since the optimal policy is numerically intractable in the general framework, the metrics proposed in 2.6, i.e., gap_{HP} and $\eta(\text{HP}, \text{SQ})$, are no longer well-defined. We introduce a new metric $\text{imp}(\text{IP}, \text{SP})$. In mathematical terms,

$$\text{imp}(\text{IP}, \text{SP}) = \frac{\text{LRAC}_{\text{SP}} - \text{LRAC}_{\text{IP}}}{\text{LRAC}_{\text{SP}}}.$$

This metric can be interpreted as the percentage improvement achieved in using the index policy (IP) instead of the shortest-queue policy (SP). For example, if the long-run average cost under SP is 100 while the long-run average cost under IP is 95, then using IP instead of SP achieves the improvement of 5%.

Since it is numerically prohibitive to compute the long-run average cost under both IP and SP, we use simulation to compare their performances. We describe the design of the simulation study below. We use the cost function introduced in Section 2.4.1. The expected number of arrivals per day, τ, is 15, same as where the cost function is minimized. Both IP and SP are operated under the same demand stream. We run 200 replications and each replication consists of 1000 days. At the end of each replication, we calculate the LRAC_{IP}, LRAC_{SP}, and $\text{imp}(\text{IP}, \text{SP})$. We then compute the mean and the standard deviation of the

$\mathrm{imp(IP, SP)}$ from 100 replications. We use the formula "mean$\pm 2\times$standard deviation" to obtain the 95% confidence interval (CI) of $\mathrm{imp(IP, SP)}$. The results are included in Table 2.6.

95% CI	$p_1 = 0$	$p_1 = 0.2$	$p_1 = 0.4$	$p_1 = 0.6$	$p_1 = 0.8$	$p_1 = 1$
$p_2 = 0$	0 ± 0	5.1 ± 0.9	8.9 ± 1.4	11.0 ± 2.0	12.5 ± 2.0	14.1 ± 2.1
$p_2 = 0.2$	3.4 ± 0.7	7.8 ± 1.2	10.2 ± 1.6	11.9 ± 2.1	13.5 ± 2.0	
$p_2 = 0.4$	5.1 ± 0.9	9.2 ± 1.3	10.8 ± 1.8	12.8 ± 2.2		
$p_2 = 0.6$	6.0 ± 1.0	10.1 ± 1.8	11.4 ± 1.9			
$p_2 = 0.8$	7.3 ± 1.2	10.8 ± 1.8				
$p_2 = 1$	8.2 ± 1.2					

Table 2.6: The 95% CI of $\mathrm{imp(IP, SP)}$ (in percentage) when $\tau = 15$.

As we can see from Table 2.6, the cost improvement increases with p_1 (decreases with p_3) for any fixed p_2 and increases with p_2 (decreases with p_3) for any fixed p_1. Note p_1 (p_3) is the proportion of the type-1 (type-3) patient, the most (least) flexible type. Hence, we conclude that the cost improvement of using IP over SP is higher when the system faces a more flexible demand.

We now consider the situation where the expected number of arrivals per day is not the same as where the cost function $c(\cdot)$ is minimized. Table 2.7 and 2.8 include the 95% CI of $\mathrm{imp(IP, SP)}$ when τ is 20 and 10 respectively. As we can see from both tables, the index policy performs better than the shortest-queue policy. However, the cost improvement when the clinic is overstaffed ($\tau = 10$) is not as high as the situation when the clinic is understaffed ($\tau = 20$).

95% CI	$p_1 = 0$	$p_1 = 0.2$	$p_1 = 0.4$	$p_1 = 0.6$	$p_1 = 0.8$	$p_1 = 1$
$p_2 = 0$	0 ± 0	5.6 ± 1.0	9.1 ± 1.3	11.4 ± 1.8	12.7 ± 2.0	14.1 ± 2.2
$p_2 = 0.2$	3.5 ± 0.7	8.0 ± 1.4	10.4 ± 1.7	12.2 ± 2.1	13.7 ± 2.2	
$p_2 = 0.4$	5.5 ± 0.8	9.4 ± 1.4	10.9 ± 1.8	12.9 ± 2.0		
$p_2 = 0.6$	6.7 ± 1.0	10.5 ± 1.7	11.9 ± 1.9			
$p_2 = 0.8$	8.1 ± 1.2	11.3 ± 1.8				
$p_2 = 1$	9.1 ± 1.3					

Table 2.7: The 95% CI of $\mathrm{imp}(\mathrm{IP}, \mathrm{SP})$ (in percentage) when $\tau = 20$.

95% CI	$p_1 = 0$	$p_1 = 0.2$	$p_1 = 0.4$	$p_1 = 0.6$	$p_1 = 0.8$	$p_1 = 1$
$p_2 = 0$	0 ± 0	2.7 ± 0.6	4.4 ± 0.9	5.0 ± 1.2	5.1 ± 1.4	5.5 ± 1.4
$p_2 = 0.2$	2.1 ± 0.4	4.1 ± 0.9	4.5 ± 1.0	4.9 ± 1.2	5.4 ± 1.5	
$p_2 = 0.4$	3.0 ± 0.6	4.4 ± 0.9	4.5 ± 1.1	5.1 ± 1.3		
$p_2 = 0.6$	3.1 ± 0.7	4.6 ± 1.2	4.8 ± 1.2			
$p_2 = 0.8$	3.6 ± 0.8	4.9 ± 1.9				
$p_2 = 1$	4.0 ± 0.9					

Table 2.8: The 95% CI of $\mathrm{imp}(\mathrm{IP}, \mathrm{SP})$ (in percentage) when $\tau = 10$.

2.8.2 Three-Day Appointment Scheduling

One of the underlying assumption of this general framework is the number of arrivals per day follows a Geometric distribution. We now investigate the case where the number of arrivals per day is a random variable with a different distribution. That is to say, the arrival process is perturbed from our assumption. Instead of building a new model to address this problem, the question we would like to answer is whether the index policy still outperforms the shortest-queue policy. (Later in this section we will modify the index policy to incorporate the perturbation of the arrival process). That is to say, is the index policy a robust policy, which means that it is resilient to the violation of the arrival process assumption?

We consider the Poisson distribution with parameter λ (the mean is λ) and Negative Binomial distribution with parameters r and p (the mean is $r \cdot \frac{1-p}{p}$). If $r = 3, p = \frac{1}{6}$ then the the expected number of arrivals per day is 15. Note that the Negative Binomial distribution is reduced to the Geometric distribution when $r = 1$ and is reduced to the Poisson distribution when $r = \infty$. Hence it can be regarded as the one between two extremes.

We include the simulation results for Negative Binomial distribution in Table 2.9 and the results for Poisson distribution in Table 2.10. Note that our observation of "more flexibility leads to higher improvement" still holds when the distribution is Negative Binomial, but it does not hold for Poisson distribution. Comparing the results in Tables 2.9 and 2.10 with those of Table 2.6, we conclude that the cost improvement under Negative Binomial and Poisson distribution are both higher than the cost improvement under the Geometric distribution.

95% CI	$p_1 = 0$	$p_1 = 0.2$	$p_1 = 0.4$	$p_1 = 0.6$	$p_1 = 0.8$	$p_1 = 1$
$p_2 = 0$	0 ± 0	11.1 ± 1.6	16.5 ± 2.0	18.0 ± 2.1	18.5 ± 2.3	18.9 ± 2.1
$p_2 = 0.2$	7.3 ± 1.1	14.9 ± 1.8	17.1 ± 2.2	18.0 ± 2.2	18.9 ± 2.3	
$p_2 = 0.4$	10.8 ± 1.6	15.8 ± 2.1	17.2 ± 2.0	18.4 ± 2.2		
$p_2 = 0.6$	11.3 ± 1.3	16.5 ± 2.2	17.5 ± 2.2			
$p_2 = 0.8$	12.4 ± 1.4	17.0 ± 2.0				
$p_2 = 1$	13.0 ± 1.5					

Table 2.9: The 95% CI of $\mathrm{imp}(\mathrm{IP}, \mathrm{SP})$ (in percentage) for Negative Binomial.

95% CI	$p_1 = 0$	$p_1 = 0.2$	$p_1 = 0.4$	$p_1 = 0.6$	$p_1 = 0.8$	$p_1 = 1$
$p_2 = 0$	0 ± 0	17.4 ± 1.9	16.8 ± 1.8	14.3 ± 1.9	12.4 ± 1.9	11.3 ± 1.9
$p_2 = 0.2$	11.7 ± 1.6	16.7 ± 1.8	14.4 ± 1.9	12.8 ± 1.9	11.7 ± 2.0	
$p_2 = 0.4$	13.9 ± 1.7	13.6 ± 2.1	12.9 ± 1.9	12.0 ± 1.8		
$p_2 = 0.6$	12.6 ± 1.5	12.3 ± 1.8	12.3 ± 1.9			
$p_2 = 0.8$	11.7 ± 1.4	11.7 ± 1.9				
$p_2 = 1$	11.0 ± 1.4					

Table 2.10: The 95% CI of $\mathrm{imp}(\mathrm{IP}, \mathrm{SP})$ (in percentage) for Poisson.

We now investigate the root cause of the observations found in Tables 2.9 and 2.10. Figure 2.5 displays the probability density functions of Geometric distribution with parameter $\alpha = \frac{15}{16}$, the Poisson distribution with parameter $\lambda = 15$ and the Negative Binomial distribution with parameters $r = 3, p = \frac{1}{6}$. Note that they all have mean 15, where the cost function is minimized. However, Poisson distribution has more density around 15, compared to the other two. If there are around 15 arrivals on most days, then the long-run average cost is supposed to be lower. This intuition is quantified in Table 2.11, which displays 95% confidence interval of the long-run average cost under three different arrival processes when $p_1 = 0.4, p_2 = 0.4, p_3 = 0.2$.

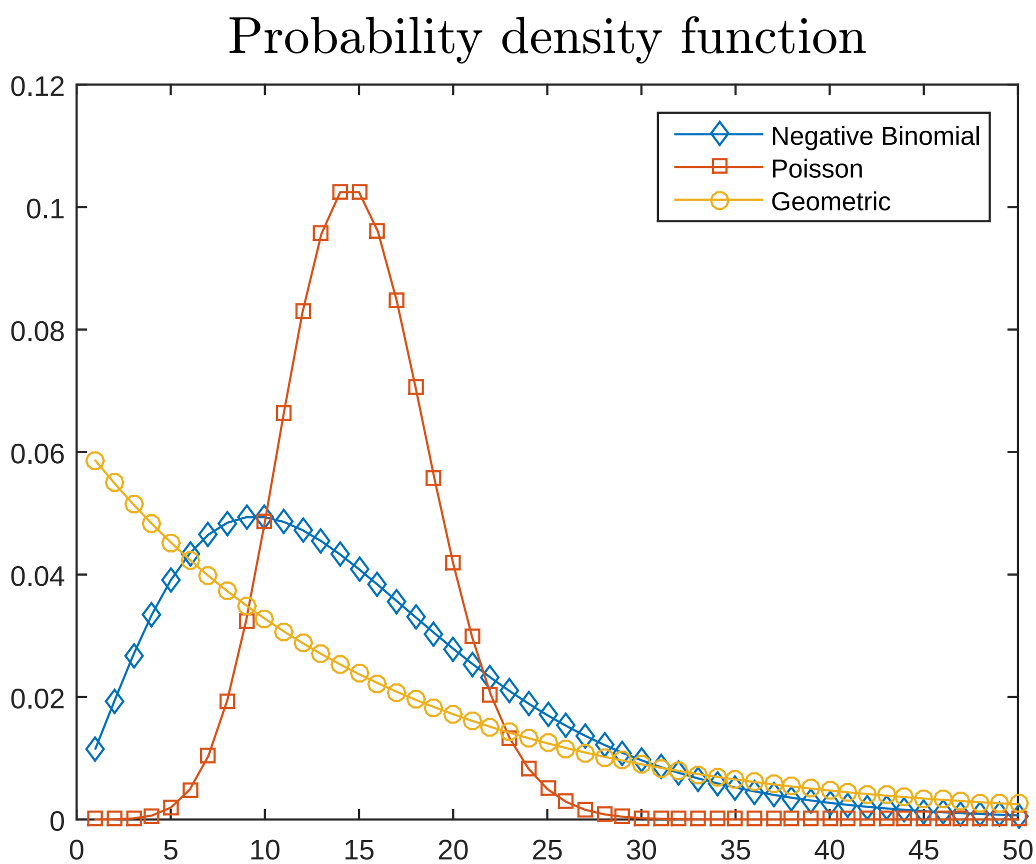

Figure 2.5: The comparison among probability density functions

As we can see Table 2.11, the probability of a Poisson random variable with mean 15 being between 10 and 20 is 0.84, much higher than that of Negative Binomial (0.44) and Geometric (0.27). This leads the fact that the magnitude of the long-run average cost under both

	$P(10 \leq X \leq 20)$	$\mathrm{LRAC}_{\mathrm{IP}}$	$\mathrm{LRAC}_{\mathrm{SP}}$
Geometric	0.27	103.3 ± 10.4	115.9 ± 10.3
Negative Binomial	0.44	58.7 ± 5.7	70.8 ± 6.2
Poisson	0.84	29.6 ± 1.4	34.1 ± 1.9

Table 2.11: The 95% CI for long-run average when $p_1 = 0.4, p_2 = 0.4, p_3 = 0.2$.

IP and SP vary under the different distributions. This results in the differences in the cost improvement of using IP over SP under the various distributions. For example, under the Poisson distribution, $\mathrm{LRAC}_{\mathrm{IP}} = 29.6$ and $\mathrm{LRAC}_{\mathrm{SP}} = 34.1$, which give $\mathrm{imp}(\mathrm{IP}, \mathrm{SP}) = 13.2\%$ while under the Geometric distribution, $\mathrm{LRAC}_{\mathrm{IP}} = 103.3$ and $\mathrm{LRAC}_{\mathrm{SP}} = 115.9$, which give $\mathrm{imp}(\mathrm{IP}, \mathrm{SP}) = 10.9\%$. These can explain why the magnitudes of cost improvements are higher in the Negative Binomial and Poisson cases.

To incorporate the perturbation of the arrival process into the index policy, we design a new policy called Poisson index policy. The computations of indices under the Poisson index policy are the same as those of the index policy, except that $A_{n,t}$ follows the distribution of $\mathrm{Poisson}(\rho_t \lambda)$. (This is due to the Bernoulli splitting of Poisson distribution.) Hence, it is easy to implement the Poisson index policy (PIP). We use $\mathrm{imp}(\mathrm{PIP}, \mathrm{SP})$ to denote the cost improvement of using Poisson index policy instead of the shortest-queue policy. Table 2.12 includes the 95% confidence interval of $\mathrm{imp}(\mathrm{PIP}, \mathrm{SP})$ when the number of arrivals per day has a Poisson(15) distribution.

Comparing Table 2.12 with Table 2.10, we found that $\mathrm{imp}(\mathrm{PIP}, \mathrm{SP})$ is in general lower than $\mathrm{imp}(\mathrm{IP}, \mathrm{SP})$, even when the number of arrivals per day has a Poisson distribution. This implies that our index policy is quite robust and performs very well.

However, it still remains unclear to us why the trend “more flexibility leads to higher

95% CI	$p_1 = 0$	$p_1 = 0.2$	$p_1 = 0.4$	$p_1 = 0.6$	$p_1 = 0.8$	$p_1 = 1$
$p_2 = 0$	0 ± 0	12.6 ± 2.2	11.6 ± 2.2	9.3 ± 2.0	8.8 ± 1.9	7.5 ± 1.9
$p_2 = 0.2$	3.5 ± 2.2	12.3 ± 2.5	9.2 ± 1.9	9.0 ± 2.1	7.9 ± 1.9	
$p_2 = 0.4$	8.6 ± 2.4	5.9 ± 2.7	8.5 ± 2.1	8.1 ± 1.9		
$p_2 = 0.6$	8.0 ± 1.8	4.9 ± 2.2	7.4 ± 2.1			
$p_2 = 0.8$	6.8 ± 1.9	4.4 ± 2.5				
$p_2 = 1$	6.2 ± 1.8					

Table 2.12: The 95% CI of $\mathrm{imp}(\mathrm{PIP}, \mathrm{SP})$ (in percentage) for Poisson(15) arrivals.

improvement" is missing when the number of arrivals per day has a Poisson distribution. We suspect that it is related to the fact that the expected number of arrivals, τ, is the same as where the cost function $c(\cdot)$ is minimized. Hence we further investigate this issue by considering the situations where τ is not 15. Table 2.13 and 2.14 include the 95% CI of $\mathrm{imp}(\mathrm{PIP}, \mathrm{SP})$ when the number of arrivals per day has a Poisson(20) and Poisson(10) distribution respectively.

95% CI	$p_1 = 0$	$p_1 = 0.2$	$p_1 = 0.4$	$p_1 = 0.6$	$p_1 = 0.8$	$p_1 = 1$
$p_2 = 0$	0 ± 0	2.5 ± 0.8	1.6 ± 0.6	1.3 ± 0.5	1.0 ± 0.4	1.0 ± 0.4
$p_2 = 0.2$	1.1 ± 1.1	1.9 ± 0.7	1.3 ± 0.6	1.0 ± 0.4	1.0 ± 0.4	
$p_2 = 0.4$	1.9 ± 0.8	0.9 ± 0.7	0.9 ± 0.5	1.0 ± 0.4		
$p_2 = 0.6$	1.7 ± 0.7	0.8 ± 0.6	0.9 ± 0.5			
$p_2 = 0.8$	1.5 ± 0.6	0.8 ± 0.5				
$p_2 = 1$	1.3 ± 0.6					

Table 2.13: The 95% CI of $\mathrm{imp}(\mathrm{PIP}, \mathrm{SP})$ (in percentage) for Poisson(20) arrivals.

95% CI	$p_1 = 0$	$p_1 = 0.2$	$p_1 = 0.4$	$p_1 = 0.6$	$p_1 = 0.8$	$p_1 = 1$
$p_2 = 0$	0 ± 0	2.1 ± 0.6	1.7 ± 0.5	0.9 ± 0.4	0.7 ± 0.3	0.6 ± 0.3
$p_2 = 0.2$	0.9 ± 0.5	1.7 ± 0.5	1.0 ± 0.4	0.7 ± 0.3	0.6 ± 0.3	
$p_2 = 0.4$	1.2 ± 0.5	0.8 ± 0.4	0.7 ± 0.3	0.6 ± 0.3		
$p_2 = 0.6$	1.0 ± 0.4	0.6 ± 0.3	0.6 ± 0.3			
$p_2 = 0.8$	0.8 ± 0.3	0.5 ± 0.3				
$p_2 = 1$	0.6 ± 0.3					

Table 2.14: The 95% CI of $\mathrm{imp}(\mathrm{PIP}, \mathrm{SP})$ (in percentage) for Poisson(10) arrivals.

As we can see from Tables 2.13 and 2.14, the Poisson index policy still outperforms the shortest-queue policy. However, the magnitude of cost improvement is much lower, compared to what we see in Table 2.12. We consider this is due to the fact that Poisson distribution has a large density condensed around its mean. We also see that the cost improvement is higher when the system is understaffed ($\tau = 20$) than overstaffed case ($\tau = 10$). This is consistent with what we have seen in Section 2.8.1. Note that “more flexibility leads to higher improvement” is still missing from both tables. By now, we have shown that this observation is due to neither the index policy with Poisson arrivals nor the inconsistency between arrivals and staffing ($\tau \neq 15$). Hence the best guess is due to the magnitude of long-run average cost under Poisson arrivals observed in Table 2.11.

Based on the observations in this section, we conclude that the index policy is a robust policy and performs quite well even if the number of arrivals per day does not have a Geometric distribution. In next section, we propose two variants of the index policy as benchmarks, to further validate the fact that our index policy is a good heuristic in general.

2.8.3 Two Variants of Index Policy

We now propose two variants of the index policy. These two variants deviate from the index policy only at the values of ρ's used to compute the indices. In the original form of the index policy, ρ's are computed by solving a quadratic programming to strike a balance among each day. We present two other ways to choose the values of ρ's, which result in two variants of the index policies.

The first variant is called “index policy - first day” (IPFD). We choose $\rho_k = p_k$ for $k = 1, 2, 3$. This index policy is improved based upon the randomized policy where all patients are given the first day in their preferred set, i.e., type-k patient arriving on day n is assigned an appointment on day $n+k$. The second variant is called "index policy - evenly assigned" (IPEA). We choose $\rho_1 = \frac{1}{3}p_1, \rho_2 = \frac{1}{3}p_1 + \frac{1}{2}p_2, \rho_3 = \frac{1}{3}p_1 + \frac{1}{2}p_2 + p_3$. This index policy is improved based upon the randomized policy where the patients are evenly assigned to each day in their preferred set. For example, $\frac{1}{3}$ of type-1 patients are assigned on day 1, 2, and 3 respectively.

We now implement these two variants of the index policy. The implementation is the same as before, except how the values of ρ's are chosen. Figure 2.6 displays the long-run average cost under IP, IPFD, IPEA, and SP across 200 replications when $p_1 = 0.1, p_2 = 0.8, p_3 = 0.1$ and the number of arrivals per day has a Geometrical distribution with mean 15.

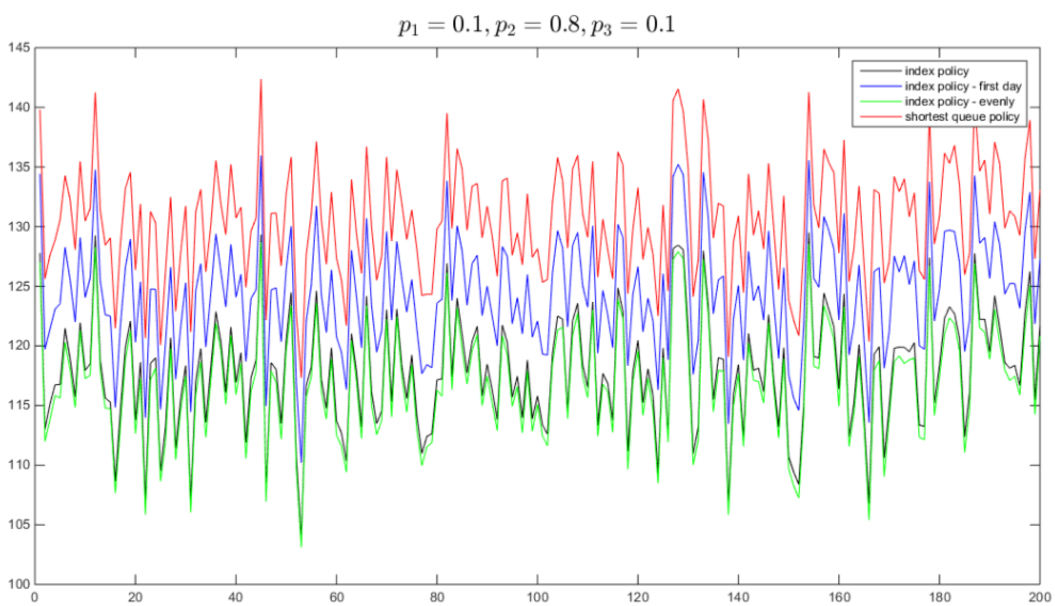

Figure 2.6: The comparisons among IP, IPFD, IPEA, and SP.

As we can see from Figure 2.6, the long-run average cost under SP is highest in all sample paths. IP, IPFD, and IPEA all outperform SP by a significant portion. We can also see that both IP and IPEA outperform IPFD. However, it is hard to see which of IP and IPEA performs better by simply eyeballing. Hence, we conduct the simulation study under more sets of parameters. Table 2.15 includes the 95% confidence interval of the difference between long-run average costs under IPFD and IP, i.e., $\mathrm{LRAC}_{\mathrm{IPFD}} - \mathrm{LRAC}_{\mathrm{IP}}$, and the difference between long-run average costs under IP and IPEA, i.e., $\mathrm{LRAC}_{\mathrm{IP}} - \mathrm{LRAC}_{\mathrm{IPEA}}$. The results are from 200 replications when the number of arrivals per day has a Geometrical distribution with mean 15.

	$\mathrm{LRAC}_{\mathrm{IPFD}} - \mathrm{LRAC}_{\mathrm{IP}}$	$\mathrm{LRAC}_{\mathrm{IP}} - \mathrm{LRAC}_{\mathrm{IPEA}}$
$p_1 = 0.1, p_2 = 0.8$	6.4 ± 1.2	0.9 ± 0.4
$p_1 = 0.2, p_2 = 0.1$	0.3 ± 0.2	0.1 ± 0.2
$p_1 = 0.2, p_2 = 0.6$	2.0 ± 0.5	1.0 ± 0.5
$p_1 = 0.3, p_2 = 0$	1.6 ± 0.5	-0.2 ± 0.2
$p_1 = 0.3, p_2 = 0.4$	0.2 ± 0.2	1.0 ± 0.5
$p_1 = 0.4, p_2 = 0.4$	1.1 ± 0.4	1.3 ± 0.5
$p_1 = 0.5, p_2 = 0$	4.4 ± 0.9	-0.7 ± 0.3
$p_1 = 0.5, p_2 = 0.4$	3.0 ± 0.7	1.5 ± 0.5
$p_1 = 0.6, p_2 = 0.2$	4.4 ± 0.9	0.6 ± 0.4
$p_1 = 0.7, p_2 = 0$	7.6 ± 1.4	-0.6 ± 0.4
$p_1 = 0.8, p_2 = 0.1$	10.4 ± 1.4	0.4 ± 0.3

Table 2.15: Comparisons among IP, IPFD, IPEA when $\tau = 15$.

As one can see from Figure 2.6, IP performs better than IPFD in all cases while one of IPEA and IP performs better than the other in various cases. Hence we conclude that both IP and IPEA outperform IPFD and IP has a similar performance as IPEA. Tables 2.16 and 2.17 display the similar results as Table 2.15 with τ being 10 and 20 respectively. We observe that the conclusion still holds in both tables.

We further study the robustness of the two variants. We consider the case where the number of arrivals per day has a Poisson distribution. We implement the two variants and

	$\mathrm{LRAC}_{\mathrm{IPFD}} - \mathrm{LRAC}_{\mathrm{IP}}$	$\mathrm{LRAC}_{\mathrm{IP}} - \mathrm{LRAC}_{\mathrm{IPEA}}$
$p_1 = 0.1, p_2 = 0.8$	4.1 ± 0.9	0.6 ± 0.3
$p_1 = 0.2, p_2 = 0.1$	0.1 ± 0.2	0.1 ± 0.2
$p_1 = 0.2, p_2 = 0.6$	1.4 ± 0.4	0.7 ± 0.5
$p_1 = 0.3, p_2 = 0$	0.7 ± 0.4	-0.1 ± 0.2
$p_1 = 0.3, p_2 = 0.4$	0.1 ± 0.2	0.7 ± 0.4
$p_1 = 0.4, p_2 = 0.4$	0.5 ± 0.3	1.0 ± 0.6
$p_1 = 0.5, p_2 = 0$	2.3 ± 0.7	-0.2 ± 0.2
$p_1 = 0.5, p_2 = 0.4$	1.6 ± 0.6	1.1 ± 0.6
$p_1 = 0.6, p_2 = 0.2$	3.1 ± 0.8	0.6 ± 0.4
$p_1 = 0.7, p_2 = 0$	4.5 ± 1.2	-0.1 ± 0.3
$p_1 = 0.8, p_2 = 0.1$	5.9 ± 1.4	0.5 ± 0.3

Table 2.16: Comparisons among IP, IPFD, IPEA when $\tau = 10$.

	$\mathrm{LRAC}_{\mathrm{IPFD}} - \mathrm{LRAC}_{\mathrm{IP}}$	$\mathrm{LRAC}_{\mathrm{IP}} - \mathrm{LRAC}_{\mathrm{IPEA}}$
$p_1 = 0.1, p_2 = 0.8$	6.4 ± 1.1	0.8 ± 0.3
$p_1 = 0.2, p_2 = 0.1$	0.4 ± 0.2	0.1 ± 0.2
$p_1 = 0.2, p_2 = 0.6$	2.1 ± 0.5	0.8 ± 0.4
$p_1 = 0.3, p_2 = 0$	2.0 ± 0.5	-0.4 ± 0.3
$p_1 = 0.3, p_2 = 0.4$	0.1 ± 0.2	0.9 ± 0.4
$p_1 = 0.4, p_2 = 0.4$	0.8 ± 0.4	1.2 ± 0.5
$p_1 = 0.5, p_2 = 0$	4.7 ± 0.9	-0.8 ± 0.4
$p_1 = 0.5, p_2 = 0.4$	2.9 ± 0.7	1.3 ± 0.5
$p_1 = 0.6, p_2 = 0.2$	3.5 ± 0.8	0.6 ± 0.3
$p_1 = 0.7, p_2 = 0$	7.3 ± 1.3	-0.6 ± 0.4
$p_1 = 0.8, p_2 = 0.1$	9.2 ± 1.6	0.5 ± 0.3

Table 2.17: Comparisons among IP, IPFD, IPEA when $\tau = 20$.

the index policy under this arrival process. Table 2.18 includes the results for Poisson(15) distribution. As we can see from Table 2.18, IP and IPEA still outperform IPFD. However, the magnitude of the performance differences is lower than what we have seen in Table 2.15.

We also consider the cases when the system is overstaffed ($\tau = 10$) or understaffed ($\tau = 20$). Table 2.19 and 2.20 include the results for Poisson(10) and Poisson(20) distribution respectively.

From Tables 2.19 and 2.20, we observe that the performance differences among IP, IPEA

	$\mathrm{LRAC_{IPFD}} - \mathrm{LRAC_{IP}}$	$\mathrm{LRAC_{IP}} - \mathrm{LRAC_{IPEA}}$
$p_1 = 0.1, p_2 = 0.8$	1.6 ± 0.4	0.6 ± 0.2
$p_1 = 0.2, p_2 = 0.1$	0.1 ± 0.2	0.6 ± 0.3
$p_1 = 0.2, p_2 = 0.6$	0.7 ± 0.3	1.3 ± 0.3
$p_1 = 0.3, p_2 = 0$	0.7 ± 0.3	0.4 ± 0.2
$p_1 = 0.3, p_2 = 0.4$	0.1 ± 0.1	1.2 ± 0.3
$p_1 = 0.4, p_2 = 0.4$	0.7 ± 0.2	0.8 ± 0.2
$p_1 = 0.5, p_2 = 0$	1.9 ± 0.4	-0.3 ± 0.2
$p_1 = 0.5, p_2 = 0.4$	0.9 ± 0.2	0.6 ± 0.2
$p_1 = 0.6, p_2 = 0.2$	1.5 ± 0.3	0.3 ± 0.1
$p_1 = 0.7, p_2 = 0$	2.4 ± 0.5	-0.2 ± 0.1
$p_1 = 0.8, p_2 = 0.1$	2.2 ± 0.5	0.3 ± 0.1

Table 2.18: Comparisons among IP, IPFD, IPEA for Poisson(15).

	$\mathrm{LRAC_{IPFD}} - \mathrm{LRAC_{IP}}$	$\mathrm{LRAC_{IP}} - \mathrm{LRAC_{IPEA}}$
$p_1 = 0.1, p_2 = 0.8$	0.8 ± 0.3	-0.1 ± 0.1
$p_1 = 0.2, p_2 = 0.1$	-0.1 ± 0.1	-0.2 ± 0.1
$p_1 = 0.2, p_2 = 0.6$	0.5 ± 0.2	-0.3 ± 0.2
$p_1 = 0.3, p_2 = 0$	-0.1 ± 0.2	-0.1 ± 0.1
$p_1 = 0.3, p_2 = 0.4$	0 ± 0.1	-0.2 ± 0.1
$p_1 = 0.4, p_2 = 0.4$	0.2 ± 0.1	-0.3 ± 0.1
$p_1 = 0.5, p_2 = 0$	0.2 ± 0.2	-0.2 ± 0.1
$p_1 = 0.5, p_2 = 0.4$	0.4 ± 0.1	-0.01 ± 0.1
$p_1 = 0.6, p_2 = 0.2$	0.6 ± 0.2	-0.04 ± 0.1
$p_1 = 0.7, p_2 = 0$	0.7 ± 0.2	-0.1 ± 0.1
$p_1 = 0.8, p_2 = 0.1$	0.6 ± 0.2	0.1 ± 0.1

Table 2.19: Comparisons among IP, IPFD, IPEA for Poisson(10).

and IPFD are very small. This is consistent with our finding in Section 2.8.2, since the Poisson distribution has much density condensed around its mean and the magnitude of the long-run average cost is much lower. Note that IP, IPEA, IPFD all outperform SP in aforementioned cases. However, we did not include the results to save some space.

Based on our observations above, we conclude that the index policy is a good heuristic policy. IP not only outperforms SP, but also outperforms IPFD. IP has a similar performance compared to IPEA, since both policies have its own advantages in different cases.

	$\mathrm{LRAC}_{\mathrm{IPFD}} - \mathrm{LRAC}_{\mathrm{IP}}$	$\mathrm{LRAC}_{\mathrm{IP}} - \mathrm{LRAC}_{\mathrm{IPEA}}$
$p_1 = 0.1, p_2 = 0.8$	0.2 ± 0.1	0.04 ± 0.05
$p_1 = 0.2, p_2 = 0.1$	0.1 ± 0.1	0.08 ± 0.06
$p_1 = 0.2, p_2 = 0.6$	0.1 ± 0.1	0.05 ± 0.06
$p_1 = 0.3, p_2 = 0$	0.1 ± 0.1	0.02 ± 0.05
$p_1 = 0.3, p_2 = 0.4$	0 ± 0	0.04 ± 0.05
$p_1 = 0.4, p_2 = 0.4$	0.1 ± 0	0.04 ± 0.05
$p_1 = 0.5, p_2 = 0$	0.3 ± 0.1	0.003 ± 0.04
$p_1 = 0.5, p_2 = 0.4$	0.1 ± 0.1	0.04 ± 0.05
$p_1 = 0.6, p_2 = 0.2$	0.2 ± 0.1	-0.01 ± 0.03
$p_1 = 0.7, p_2 = 0$	0.4 ± 0.1	-0.05 ± 0.05
$p_1 = 0.8, p_2 = 0.1$	0.3 ± 0.1	0.01 ± 0.03

Table 2.20: Comparisons among IP, IPFD, IPEA for Poisson(20).

2.9 Conclusions

In this chapter, we have developed a novel way of modeling patient preferences via the set of days that the patient is indifferent to. We developed a dynamic appointment scheduling system to help clinics respond to the patients' requests by assigning the appointments in a cost-effective way.

We used MDPs to obtain optimal policies that minimize the long-run average cost. We presented analytical results about the structure of the optimal policy. We showed interesting structural properties of the optimal policies. Since the optimal policy is hard to implement, we proposed a heuristic policy, namely, the index policy. The numerical results indicate that the index policy performs very close to the optimal policy and provides significant improvement over the other heuristic policies. We extended the results from the base model to the case where the rejection is available. We also extended the results to a general framework and used a simple case to study the generalizations of arrival processes.

CHAPTER 3: Automated Teller Machines Replenishment Policies

3.1 Introduction

Automated teller machines (ATMs) can potentially save costs (including labor and capital) for banks and improve customer service. Replenishing the ATMs costs money, so the banks would like to do it as infrequently as possible. However, customer satisfaction is directly dependent on finding ATMs with cash. This will increase by replenishing the ATMs more frequently. Thus the banks need to strike a balance between these two competing factors to decide upon an intelligent replenishment policy.

With the recent advances in technology, the banks are able to know the cash inventory at all ATMs in real time. With the help of sophisticated statistical analysis, the banks can forecast the cash demands accurately. The banks can exploit such information to make decisions on which ATMs to replenish. There are two major ways the banks handle the replenishment options. Under option one, they outsource the replenishment operations to an outside vendor, while under option two, they own the ATMs and perform the replenishment operations in house. We discuss these two cases below.

- **Outsourcing operations**: Under this scenario, the banks have a replenishment contract with a vendor. The outsourcing contract usually includes a certain amount of fee for each replenishment (variable cost) and the regular annual contract allowance (fixed cost, which may be zero). In this situation, the replenishment cost function is additive, that is, the cost of replenishing k ATMs is linear in k.

- **In-house operations**: Under this scenario, the banks manage the cash replenishment operations themselves. Here we distinguish between on-site ATMs (that is, ATMs that are in a bank building) and off-site ATMs (that is, ATMs that are in shoppings, train and bus stations, etc). We concentrate on the off-site ATMs. In this case, it is more cost-effective to replenish multiple ATMs simultaneously, due to the economies of scale. This typically makes the replenishment cost a submodular function of the set of ATMs that are replenished.

In reality, some customers may deposit cash into ATMs but we ignore deposits in this model. Right before each time period starts, the banks observe the current cash inventory level at each ATM. (In this model, we use the day as the time period.) If there is no cash in a certain ATM, then the stock-out cost is incurred. This stock-out cost can be explained as either an estimate of good will loss or the emergency replenishment cost. The replenishments of ATMs occur instantaneously. This is because the data we have show that customer cash withdrawals are negligible during 2-4AM each day. Therefore, we can assume the replenishment happens during this time interval. We begin with the assumption that the cash withdrawals at each ATM are independent and identically distributed, and independent of the demands at other ATMs. (However, we allow the demand distributions to depend on the ATM.) This may not actually be true in practice, but the assumption is useful in developing tractable policies. Also, as we shall see in Section 3.9, the tractable policies do quite well in the presence of real world data, which may have auto- and cross-correlated demands. We do not model the cash withdrawal of each customer explicitly. Instead, we assume the cumulative cash withdrawals during a day have a given probability distribution which is estimated based on the historical data. Although, for mathematical tractability, our model assumes that the amounts of cash withdrawals during each day at each ATM are independent and identically distributed, the actual index policy we propose does not need this assumption.

We address two issues related to the replenishment operations below. The first issue is how the replenishment operation is completed. Each ATM has a sealed cartridge of cash. The trucks with full sealed cartridges of cash stop at each ATM and the operators replace the sealed cartridge in ATM (which is now partially or fully empty of cash) with the sealed cartridge full of cash. If there is leftover cash in the cartridges, it will be transported to the central location and processed there. This process ensures that the operators have no contact with the cash for the security reasons.

The second issue is the banks' ability to provide enough cash and put them into the sealed cartridges for future use. For example, the banks might need to replenish 20 ATMs on one day but they have only 10 full cartridges available. We do not consider this second factor in the main model but discuss it in the last section. Here we assume that enough full cartridges are available to replenish as many ATMs as needed.

We build a Markov Decision Processes (MDP) model to construct the replenishment schedules, taking all the factors mentioned above into account. We consider two costs: the replenishment costs and the stock-out costs. The replenishment costs are typically easy to estimate, but stock-out costs are harder to do so. The stock-out costs will induce a certain replenishment policy, which induces a certain stock-out frequency, which is an indicator of the service level. Hence, we can control the service level by choosing appropriate stock-out costs. The MDP model is designed to help the bank decide which ATMs to replenish at the beginning of each day. The objective is to minimize the long-run average stock-out costs plus replenishment costs for the banks.

Our work makes several contributions. The first contribution is the construction of an MDP model to study the replenishment scheduling of ATMs. To the best of our knowledge, this is the first MDP model in the literature on ATMs replenishment. The second contribution is to apply the greedy algorithm to choose the heuristic policy. Many MDP models in previous literature have large state spaces which make the optimal solution analytically intractable. Our MDP has a large state space, as well as a large action space (the power set of the set of ATMs). This makes our problems distinct from the previous MDP models. Finally, although we regard the ATM replenishment problem as the business context here, the methodology and approaches developed here can be easily applied to broader business settings, such as businesses owning and operating vending machines.

The rest of Chapter 3 is organized as follows. The related literature is reviewed in Section 3.2. We describe the MDP model in detail in Section 3.3. The simplest case, a single-ATM problem, is reviewed in Section 3.4. A multi-ATM problem with additive replenishment cost is fully solved in Section 3.5. We then show the structural properties of the optimal policy (with a general cost function) in Section 3.6, propose the index policy in Section 3.7, and report numerical experiments in Section 3.8 to show that the index policy performs well with respect to the optimal policy when it can be computed, and with respect to another benchmark policy when the optimal policy is hard to compute. The statistical analysis and the performance of the index policy on a real-world dataset are presented in Section 3.9. Finally, we conclude this work and list several future directions in Section 3.10.

3.2 Literature Review

The replenishment of a single product has been well studied in the classical inventory control literature, dating back to the Economic Order Quantity (EOQ) model proposed in (Harris, 1990). (Arrow et al., 1951) and (Clark and Scarf, 1960) studied the optimal order quantity in a single-echelon case and a multi-echelon case respectively. (Scarf, 1959) and (Iglehart, 1963) have shown the optimality of the (s, S) policy under the total discounted cost criterion in a finite-horizon case and an infinite-horizon case respectively. (Veinott Jr and Wagner, 1965) have shown the optimality of the same policy under the average-cost criterion. We refer the reader to (Porteus, 1990) for a review on stochastic inventory control literature.

The replenishment of multiple products at a single location is related to the joint replenishment problem (JRP) literature. JRP also includes the problems of single-product replenishment at multiple locations. We refer the reader to the review on the literature prior to 1989 in (Goyal and Satir, 1989) and the review on the literature between 1989 and 2005 in (Khouja and Goyal, 2008). The (s, c, S) policy, often called coordinated control policy, was originally proposed in (Balintfy, 1964), with S being the *order-up-to* level, c being *can-order* level, and s being *must-order* level. An order is placed when the inventory position of any item is below the must-order level. The fulfillment of order raises the inventory position of this item to the order-up-to level. When an order is placed, any other item with an inventory position below the can-order level is also included in the order. Note that the (s, S) policy in a multi-item setting can be regarded as a special case of the (s, c, S) policy, with the can-order level being the same as the must-order level. Although (Ignall, 1969) showed that (s, c, S) policy is not necessarily optimal, such policies have been studied extensively in the JRP literature when the demand is random. The literature on JRP has mostly focused on the problems arising in a single-location-multiple-product setting, where the ordering cost consists of two parts: the major ordering cost associated with each order and the minor ordering cost associated with each product or part. Hence the ordering cost is an affine function of the number of products or parts, e.g. see (Federgruen et al., 1984). (Federgruen et al., 1984) explicitly took the service level constraint into consideration, set the required service level and solved the problem using the Lagrangian Multiplier method. They considered the following cost structure: a major setup cost incurred by any replenishment and a minor setup cost when including a particular item in the replenishment. They searched for a (s, c, S) policy to control the multi-location inventory systems for coordinated deliveries. Our model differs in the following way: we consider which ATMs to replenish as the decision variable and model the cost structure as a submodular function.

Most of the literature on JRP considers the ordering cost and inventory holding cost, without taking the transportation cost into account explicitly. Although the ordering cost reflects the transportation cost indirectly, most of JRP literature ignores the routing problem. There is a stream of literature which began in 1980s, called the inventory routing problems (IRP), which considers both inventory cost and distribution(transportation) cost. Many IRP arise from a relatively new business model called vendor managed inventory (VMI) where the supplier reviews the clients' inventory levels and replenishes the inventories for them, minimizing the distribution and inventory costs throughout the supply chain. We refer the reader to (Federgruen and Simchi-Levi, 1995) and (Cordeau et al., 2007) for the reviews on IRP. (Federgruen and Zipkin, 1984), one of the pioneering papers in the IRP literature, is the first to propose an integrated inventory management and transportation planning model. They consider inventory cost, shortage cost due to random demand, as well as transportation cost. They extend the standard vehicle routing problems (VRP) and solve the problem using an algorithm derived from generalized Bender's decomposition. Their model considers only one time period and hence solves a "myopic version of the problem", whereas our model minimizes the long-run average cost. (Burns et al., 1985) is the first paper to combine inventory cost with the transportation cost as a function of traveling salesman tour. Their model considers shipment size as the decision variable and shows that the optimal shipment size is a full truck while the distribution strategy is to dispatch trucks which deliver items to more than one customer per load. Their model assumes the demand is a known constant and uses the information on the spatial density of the retailers. Our model differs in the following ways: we consider which ATMs to replenish as the decision variable, take the demand uncertainty into consideration, and use the exact locations of the retailers. (Herer and Roundy, 1997) assumed that the ordering cost consists of two parts: the first part is a submodular function of the set of retailers placing orders, and the second part is a linear function of the traveling salesman tour length. They focused on the systems where the delivery costs are significant, such as the grocery store chain, and the delivery of the gasoline to service stations. They assumed the demand occurs at the retailers at a constant known rate, considered when to deliver products from the central warehouse to retailers and when to replenish the inventories at the central warehouse as the decision variables, and solved the problem by computing the power-of-two reorder intervals (the reorder interval is a positive or negative integer power-of-two times the base planning period). Our model differs in the following ways: we estimate the probability distribution of the demand, consider which ATMs to replenish as the decision variable, and leverage such information and MDP to design the heuristic policy. We also use submodular function in our cost model.

3.3 Model Description

Let $N = \{1, 2, ..., n\}$ denote a set of ATMs. Let M_i be the capacity of ATM $i \in N$, that is, ATM i can hold at most M_i amount of cash at any time. Let $D_k^{(i)}$ be the size of cash withdrawal that occurs at ATM i on day k. We assume that $\{D_k^{(i)}, k = 1, 2, \cdots\}$ is a sequence of iid non-negative random variables with a known common complementary cumulative distribution given by

$$f^{(i)}(y) = \mathrm{P}(D_k^{(i)} \geq y), \quad \forall y \geq 0.$$

We use $D^{(i)}$ to denote a generic random size of daily cash withdrawal at ATM i.

We model the cost structure as follows. If A is the set of ATMs to be replenished, then the bank incurs a replenishment cost $r(A)$. We consider two different cases: (1) the replenishment cost function $r(A)$ is additive and (2) the replenishment cost function $r(A)$ is submodular. In case (1), the replenishment cost is a simple sum of replenishment costs incurred at all ATMs to be replenished. That is to say, if the replenishment cost of ATM i is r_i, then

$$r(A) = \sum_{i \in A} r_i.$$

This situation occurs when the bank has a contract with third-party vendor who replenishes the ATMs for the bank and charges the bank a fixed fee for each replenishment. This is a "pay-per-replenishment" contract with $r_i = r$ for all $i \in N$. In case (2), the replenishment cost function is a submodular function, that is, it satisfies the following inequality:

$$r(A \cup B) + r(A \cap B) \leq r(A) + r(B) \quad \forall A, B \subseteq N.$$

One can show that this is equivalent to (see (Herer, 1999)):

$$r(A \cup \{i, j\}) + r(A) \leq r(A \cup \{j\}) + r(A \cup \{i\}) \quad \forall A \subseteq N,\ i, j \notin A. \tag{3.1}$$

This definition has an implication of economies of scale: it costs less to add ATM j to the replenishment set $A \cup \{i\}$ than to add ATM j to the replenishment set A.

To construct the MDP, we describe the sequence of events as follows. Let X_k be the state of the system before a potential replenishment happens on the kth day $(k \geq 1)$. The state is given by a vector of non-negative integers $X_k = (X_k^{(1)}, X_k^{(2)}, ..., X_k^{(n)})$, where $X_k^{(i)}$ is the cash inventory level of ATM i (for $i = 1, 2, ..., n$) just before the replenishment epoch on day k. The state space is denoted by

$$S = \{x = (x^{(1)}, x^{(2)}, ..., x^{(n)}) : 0 \leq x^{(i)} \leq M_i\}.$$

The stock-out cost is incurred based on the current cash inventory level. The total stock-out cost is equal to the sum of the stock-out costs incurred at each ATM. We use $c(x)$ to denote the total stock-out cost when the system state is $x = (x^{(1)}, x^{(2)}, ..., x^{(n)})$. We assume that it costs c_i if ATM i is without cash at the replenishment epoch. Thus

$$c(x) = \sum_{i=1}^{n} c_i \cdot 1_{\{x^{(i)}=0\}}. \tag{3.2}$$

The stock-out cost c_i is an intangible measure and hard to estimate. Intuitively, higher stock-out cost induces higher service level. We explain how to estimate the stock-out cost given the required service level in Section 3.4.

A decision A_k is the subset of the ATMs that will be replenished at decision epoch k. This decision has to be made before the kth day starts, incurring the replenishment cost $r(A_k)$. During the kth day, the cash withdrawals are denoted by $D_k = (D_k^{(1)}, D_k^{(2)}, ..., D_k^{(n)})$, where $D_k^{(i)}$ is the amount of cash withdrawals from ATM i during day k, for $i = 1, 2, ..., n$. These will result in a change of state of the system from X_k to $X_{k+1} = y(X_k, A_k)$, where

$$X_{k+1}^{(i)} = y^{(i)}(X_k, A_k) = \begin{cases} (M_i - D_k^{(i)})^+, & \text{if } i \in A_k, \\ (X_k^{(i)} - D_k^{(i)})^+, & \text{if } i \notin A_k, \end{cases}$$

We see that the independence assumption on the demands implies that $\{(X_k, A_k), k \geq 1\}$ is an MDP. The objective is to minimize the long-run average cost per day. For a given stationary Markovian policy π, define (assuming the limit exists)

$$g^{\pi}(x) = \lim_{k\to\infty} \frac{1}{k} \mathrm{E}^{\pi} \left[\sum_{j=1}^{k} r(A_j) + c(X_j) \middle| X_1 = x \right].$$

We call $g^{\pi}(x)$ the long-run average cost of following the policy π. Let

$$g^{*}(x) = \inf_{\pi} g^{\pi}(x), \quad \forall x \in S.$$

If there is a policy π^{*} that achieves this infimum, it is called the average-cost optimal policy. Thus an optimal policy (if it exists) satisfies

$$g^{\pi^{*}}(x) = g^{*}(x), \quad x \in S.$$

We next discuss when such an optimal policy exists and how to compute it. Define $v_0(x) = 0$ for all x and, for $k \geq 1$,

$$v_{k+1}(x) = c(x) + \min_{A \subseteq N} \{ r(A) + \mathrm{E}[v_k(y(x, A))] \}, \tag{3.3}$$

where $x = (x^{(1)}, x^{(2)}, ..., x^{(n)}) \in S$, $y(x, A) = (y^{(1)}(x, A), y^{(2)}(x, A), ..., y^{(n)}(x, A)) \in S$. Note that one can interpret $v_k(x)$ as the optimal total expected cost incurred over the k days starting from state x. It is well known (see (Tijms, 2003)) that $v_k(x)$ is asymptotically linear in k with slope g and intercept $h(x)$. More precisely, we can write

$$v_k(x) = kg + h(x) + o(k)$$

where $o(k)$ is any function such that $\frac{o(k)}{k} \to 0$ as $k \to \infty$. The slope g is the optimal long-run average cost and the intercept $h(\cdot)$ is called the bias function. It is also known (see (Tijms, 2003)) that g and $h(\cdot)$ satisfy the following Bellman equation

$$h(x) + g = c(x) + \min_{A \subseteq N} \{r(A) + \mathrm{E}[h(y(x, A))]\}, \quad \forall x \in S. \tag{3.4}$$

We also know (see (Tijms, 2003)) that if Equation (3.4) has a solution, then it can be used to compute the optimal policy as follows. Define

$$A(x) = \arg\min_{A \subseteq N} \{c(x) + r(A) + \mathrm{E}[h(y(x, A))]\}, \quad \forall x \in S. \tag{3.5}$$

The standard theory of dynamic programming (see (Tijms, 2003)) shows that the Markovian policy that replenishes the set $A(x)$ of ATMs in state x is optimal. (Tijms, 2003) also shows that Equation (3.4) has a solution if the MDP is unichain, that is, for each stationary policy the associated Markov chain has no two disjoint closed sets. Thus if we can show that our MDP is unichain, then we have a method of computing the optimal policy. This is done in the following theorem.

Theorem 12. *Suppose*

$$f_k^{(i)}(M_i) > 0, \quad \forall i = 1, 2, ..., n. \tag{3.6}$$

Then the Markov decision process $\{(X_k, A_k), k \geq 1\}$ *is unichain and aperiodic.*

Proof. We have, for any policy π,

$$\begin{aligned} & P\left(X_{k+1} = (0, 0, ..., 0) | X_k = (X_k^{(1)}, X_k^{(2)}, ..., X_k^{(n)})\right) \\ \geq & \prod_{i=1}^{n} \mathrm{P}(D_k^{(i)} \geq M_i) = \prod_{i=1}^{n} f_k^{(i)}(M_i) > 0. \end{aligned}$$

Hence the system can go from any state $(x_k^{(1)}, x_k^{(2)}, ..., x_k^{(n)})$ to state $(0, 0, ..., 0)$ in one step with a positive probability, regardless of the policy followed. Also, state $(0, 0, ..., 0)$ is aperiodic because

$$\begin{aligned} & \mathrm{P}(X_{k+1} = (0, 0, ..., 0) | X_k = (0, 0, ..., 0)) \\ \geq & \prod_{i=1}^{n} \mathrm{P}(D_k^{(i)} \geq M_i) = \prod_{i=1}^{n} f_k^{(i)}(M_i) > 0. \end{aligned}$$

Therefore $\{(X_k, A_k), k \geq 1\}$ satisfies the unichain assumption. □

The assumption in Equation (3.6) says that there is a positive probability that all the cash in an ATM can be withdrawn in one day. This is a sufficient condition, and can easily be relaxed. However, this assumption is easily met in practice, and hence we do not explore weaker sufficient conditions. We shall assume Equation (3.6) holds. Then the MDP is unichain, and hence Equation (3.4) has a solution. We solve the Equation (3.4) by the iterative method in Equation (3.3). The following theorem is an application of Theorem 6.6.1 in (Tijms, 2003). Let $\mathbf{0} = (0, 0, ..., 0)$ be the state where all ATM cash levels are zero.

Theorem 13. *For any state x, we have*

$$h(x) - h(\mathbf{0}) = \lim_{k \to \infty} [v_k(x) - v_k(\mathbf{0})], \quad \forall x \in S, \tag{3.7}$$

and

$$g = \lim_{k \to \infty} \frac{v_k(x)}{k}, \quad \forall x \in S. \tag{3.8}$$

Furthermore,

$$\min_x \{v_k(x) - v_{k-1}(x)\} \le g \le \max_x \{v_k(x) - v_{k-1}(x)\}. \tag{3.9}$$

Proof. Use the Theorem 6.6.1 in (Tijms, 2003). □

Theorem 13 allows us to use the recursion in Equation (3.3) to compute $h(x)$ and g, and then use Equation (3.5) to derive the optimal policy.

3.4 Single-ATM Problem

In this section we consider the special case $n = 1$, that is, a single-ATM problem. Let M be the capacity, c be the stock-out cost, r be the replenishment cost, and D be the stationary demand for this ATM. In this case, Equation (3.4) can be written as:

$$h(x) + g = c \cdot 1_{\{x=0\}} + \min\{r + \mathrm{E}[h((M-D)^+)], \mathrm{E}[h((x-D)^+)]\}, \tag{3.10}$$

where x is a scalar, representing the ATM cash level. This case has been studied in great detail in the literature starting with the seminal paper (Scarf, 1959), which showed that an (s, S) policy minimizes the finite-horizon expected total discounted cost. This was later extended to the infinite-horizon case by (Iglehart, 1963) considering a periodic review inventory model with ordering cost, shortage cost, and holding cost. (Veinott Jr and Wagner, 1965) proved the optimality of the (s, S) policy under the average-cost criterion. The single-ATM case is in fact a special case of (Iglehart, 1963) with a fixed quantity $S = M$, no holding cost, the replenishment cost r taking place of the ordering cost, and the stock-out cost c taking place of the shortage cost. Hence we see that the long-run average cost is minimized by an (s, M) policy for some s.

We now give the following notation. Suppose D_k is the demand at this ATM during day k. Assume that $\{D_k, k \geq 1\}$ is a sequence of independent and identically distributed random variables with

$$f(y) = \mathrm{P}(D_k \geq y), \quad \forall y \geq 0.$$

Let $\{\Lambda(t), t \geq 0\}$ be a renewal process generated by $\{D_k, k \geq 0\}$, and let $\{B(t), t \geq 0\}$ be the remaining life process in this renewal process; see (Kulkarni, 2009). The next theorem gives an expression for $\kappa(s)$, the long-run average cost of such a policy.

Theorem 14. *The long-run average cost under the* (s, M) *policy is given by:*

$$\kappa(s) = \frac{r + c\mathrm{P}(B(M-s) \geq s)}{1 + \mathrm{E}[\Lambda(M-s)]}. \tag{3.11}$$

Proof. Let W_k be the cash inventory instantly after the replenishment happens on day k. Suppose $W_0 = M$. Then under the (s, M) policy, we get

$$W_{k+1} = \begin{cases} W_k - D_k, & \text{if } W_k - D_k \geq s, \\ M, & \text{if } W_k - D_k < s. \end{cases}$$

Let T_k be the time of the kth replacement. Hence $\{W_k, k \geq 1\}$ is a regenerative process with regeneration points $\{T_k, k \geq 1\}$. We call $(0, T_1]$ as the first cycle, starting with $W_0 = M$. Then the expected cycle length is $\mathrm{E}(T_1) = 1 + \mathrm{E}[\Lambda(M-s)]$. We incur a replenishment cost of r at the end of the cycle. We incur a stock-out cost of c if the cash inventory at the end of the cycle is less than or equal to zero, that is, if the remaining life at $M - s$ in the $\{\Lambda(t), t \geq 0\}$ process is s or more. Hence the expected cost over the first cycle is given by $r + c\mathrm{P}(B(M-s) \geq s)$. From the theory of Renewed Reward Process (e.g., see Theorem 8.38 in (Kulkarni, 2009)), we have

$$\kappa(s) = \frac{r + c\mathrm{P}(B(M-s) \geq s)}{1 + \mathrm{E}[\Lambda(M-s)]}.$$

$\square$

We can compute $\mathrm{P}(B(M-s) \geq s)$ and $\mathrm{E}[\Lambda(M-s)]$ by using the renewal argument;

see (Kulkarni, 2009). Let s^* be the s that minimizes $\kappa(s)$ of Equation (3.11). In general r is easy to estimate, but c is not. We now discuss how this model can be used to pick the "right" c.

Let $\rho(s)$ be the service level, which is defined as the fraction of the days when the ATM is not empty. The following theorem gives the explicit expression of service level in this model.

Theorem 15. *The long-run average service level under the* (s, M) *policy is given by:*

$$\rho(s) = 1 - \frac{\mathrm{P}(B(M-s) \geq s)}{1 + \mathrm{E}[\Lambda(M-s)]}. \tag{3.12}$$

Proof. We follow the same notations in the proof Theorem 14. During the first cycle $(0, T_1]$, the stock-out occurs with $\mathrm{P}(B(M-s) \geq s)$ and does not occur with probability $1-\mathrm{P}(B(M-s) \geq s)$. From the theory of Regenerative Process (e.g., see Theorem 8.39 in (Kulkarni, 2009)) the expected number of stock-out is

$$\lim_{n\to\infty} \mathrm{P}(W_n \leq 0) = \frac{\mathrm{P}(B(M-s) \geq s)}{1 + \mathrm{E}[\Lambda(M-s)]}.$$

Hence by the definition of the service level, we complete the proof. $\square$

Note that the service level for a single ATM is bounded above by $1 - \mathrm{P}(D \geq M) = \mathrm{P}(D < M)$ where D denotes the daily cash demand. This is because, even if the replenishment occurs every day, the fraction of stock-out occurring in the long-run is $\mathrm{P}(D \geq M)$. Hence the service level must be less than or equal to $\mathrm{P}(D < M)$. For a given c, let $s^*(c)$ be the replenishment point which minimizes the $\kappa(s)$ of Equation (3.11). It can be shown that

$\rho(s^*(c))$ is an increasing function of c. Thus one can choose a sufficiently large c to achieve any service level $\rho(s^*(c))$ (less than $\mathrm{P}(D < M)$).

For example, Figure 3.1 displays the service level $\rho(s^*(c))$ as a function of stock-out c, given the parameters $r = 60$, Poisson(25) demands, and ATM capacity $M = 50$. We take the smallest c such that the service level is above 0.95 as the "reasonable" c, so $c = 140$.

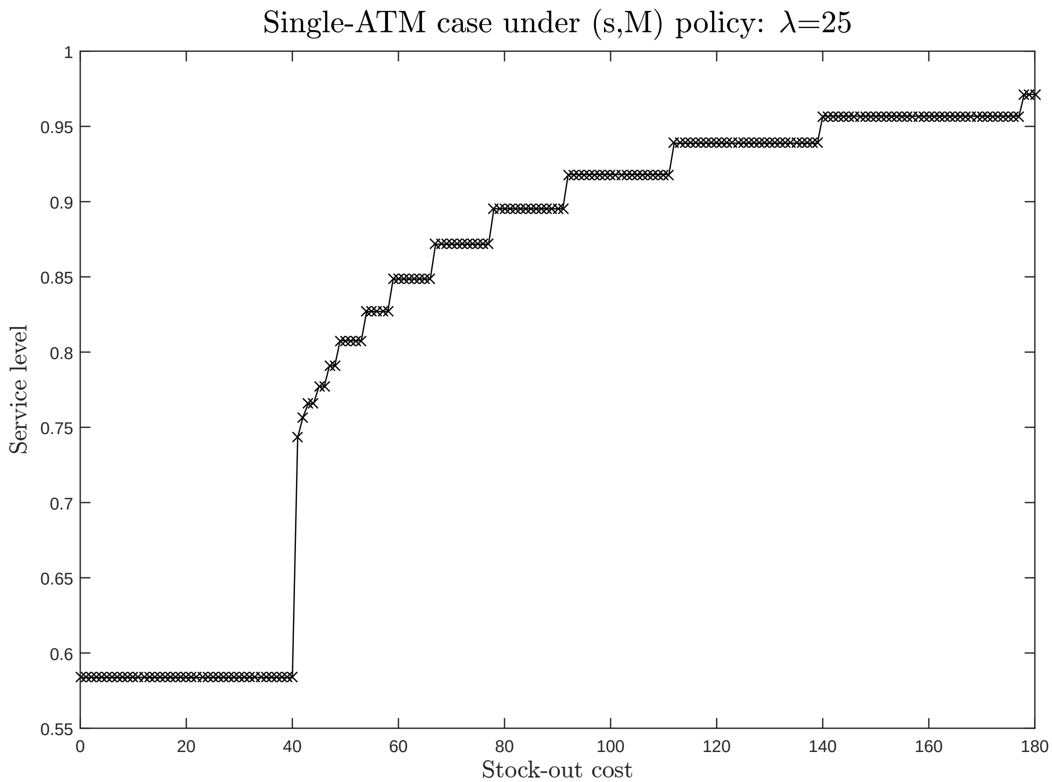

Figure 3.1: Relationship between service level and stock-out cost

3.5 Additive-Replenishment-Cost Problem

In this section we analyze the case where the replenishment cost is additive, i.e.,

$$r(A) = \sum_{i \in A} r_i.$$

It was mentioned in Section 3.1 that this cost function arises when the bank outsources the replenishment operations. Let M_i be the capacity, r_i be the replenishment cost, c_i be the stock-out cost, and $D^{(i)}$ be the stationary demand for ATM i. Let $h_i(x^{(i)})$ and g_i be solutions to Equation (3.10), rewritten here with the superscript i added to denote ATM i,

$$h_i(x^{(i)}) + g_i = c_i \cdot 1_{\{x^{(i)}=0\}} + \min\{r_i + \mathrm{E}[h_i((M_i - D^{(i)})^+)], \mathrm{E}[h_i((x^{(i)} - D^{(i)})^+)]\}. \tag{3.13}$$

The optimal replenishment level $s^{(i)}$, computed by minimizing the long-run average cost $\kappa_i(s)$, satisfies

$$g_i = \kappa_i(s^{(i)}).$$

Theorem 16. *Define*

$$h(x) = \sum_{i=1}^{n} h^{(i)}(x^{(i)}), \quad g = \sum_{i=1}^{n} g_i.$$

Then $h(x)$ and g satisfy Equation (3.4).

Proof. We solve Equations (3.13) by the following recursion, $\forall i \in N$,

$$\begin{aligned} v_{k+1}^{(i)}(x^{(i)}) =& c_i \cdot 1_{\{x^{(i)}=0\}} \\ &+ \min\{r_i + \mathrm{E}[v_k^{(i)}((M_i - D^{(i)})^+)], \mathrm{E}[v_k^{(i)}((x^{(i)} - D^{(i)})^+)]\}. \end{aligned}$$

For any non-negative quantities $U_1, V_1, U_2, V_2, ..., U_n, V_n$, we know

$$\min\{U_1, V_1\} + \min\{U_2, V_2\} + \cdots + \min\{U_n, V_n\} = \min_{A \subseteq N} \left\{ \sum_{i \in A} U_i + \sum_{i \notin A} V_i \right\}. \quad (3.14)$$

We sum all the Equations in (3.13) and use Equation (3.14) to get:

$$\begin{aligned}
&\sum_{i=1}^{n} v_{k+1}^{(i)}(x^{(i)}) \\
=&\sum_{i=1}^{n} c_i \cdot 1_{\{x^{(i)}=0\}} + \min_{A \subseteq N}\{\sum_{i=1}^{n} r_i + \sum_{i \in A} \mathrm{E}[v_k^{(i)}((M_i - D^{(i)})^+)] \\
&+ \sum_{i \notin A} \mathrm{E}[v_k^{(i)}((x^{(i)} - D^{(i)})^+)]\} \\
=&c(x) + \min_{A \subseteq N}\{r(A) + \mathrm{E}[\sum_{i \in A} v_k^{(i)}((M_i - D^{(i)})^+)] + \sum_{i \notin A} v_k^{(i)}((x^{(i)} - D^{(i)})^+)\} \\
=&c(x) + \min_{A \subseteq N}\{r(A) + \mathrm{E}[\sum_{i=1}^{n} v_k^{(i)}(y^{(i)}(x, A))]\}.
\end{aligned}$$

This is equivalent to Equation (3.3) if we define

$$v_k(x) = \sum_{i=1}^{n} v_k^{(i)}(x^{(i)}).$$

$\square$

Theorem 16 demonstrates that, in the additive-replenishment-cost case the n-ATM problem can be solved by solving n single-ATM problems separately and obtaining the optimal replenishment level $s^{(i)}$ for ATM i ($i = 1, 2, ..., n$). We can calculate $s^{(i)}$ by methods mentioned in Section 3.4. The optimal decision in the state of x is characterized by:

$$A(x) = \{i \in N : x^{(i)} < s^{(i)}\}.$$

Let $s = (s^{(1)}, ..., s^{(n)})$ and $M = (M_1, ..., M_n)$. We call such a policy an (s, M) policy. Theorem 16 implies that an (s, M) policy is optimal for some s. Thus the problem of optimal replenishment is easy if the bank is outsourcing its replenishment operations.

3.6 Structural Properties of Optimal Policy

In this section, we study the structural properties of the optimal policy for an ATM replenishment problem with a general replenishment cost function $r(A)$.

Assumption A: We assume that $c(x)$ satisfies

$$z \geq x \Rightarrow c(z) \leq c(x). \tag{3.15}$$

This assumption says the stock-out cost is lower when there is more cash in any ATM. This is a reasonable assumption in reality. The stock-out cost in Equation (3.2) satisfies Equation (3.15). We first establish the monotonicity of $h(\cdot)$ function of Equation (3.4) in the next theorem.

Theorem 17. *Under the Assumption A, we have:*

$$z \geq x \Rightarrow h(z) \leq h(x), \quad \forall x, z \in S.$$

Proof. Let $e^{(i)} = (0, ..., 0, 1, 0, ..., 0)$, where the ith component of this vector is 1 and all other components are 0. Since we can write $z = (z^{(1)}, ..., z^{(n)}) = (x^{(1)} + \epsilon_1 e^{(1)}, ..., x^{(n)} + \epsilon_n e^{(n)})$, where $\epsilon_i = z^{(i)} - x^{(i)} \geq 0$, it suffices to show that

$$h(x) \geq h(x + \epsilon e^{(i)}), \quad \forall \epsilon \geq 0, i \in N, x \in S.$$

We first show

$$v_k(x) \geq v_k(x + \epsilon e^{(i)}), \quad \forall k \geq 1, \epsilon \geq 0, i \in N, x \in S$$

by mathematical induction.

Step 1: The basic step. We know $v_0(x) = 0$. Hence we have

$$\begin{aligned} &v_1(x) = c(x) + \min_{A \subseteq N}\{r(A) + 0\} = c(x) + 0 = c(x), \\ &v_1(x + \epsilon e^{(i)}) = c(x + \epsilon e^{(i)}) + \min_{A \subseteq N}\{r(A) + 0\} = c(x + \epsilon e^{(i)}) + 0 = c(x + \epsilon e^{(i)}). \end{aligned}$$

Hence $v_1(x) \geq v_1(x + \epsilon e^{(i)})$ holds given that $c(x) \geq c(x + \epsilon e^{(i)})$.

Step 2: The inductive step. We now assume that $v_k(x) \geq v_k(x + \epsilon e^{(i)})$ for all $\epsilon \geq 0, i \in$

$N, x \in S$ and show that $v_{k+1}(x) \geq v_{k+1}(x + \epsilon e^{(i)})$ for all $\epsilon \geq 0, i \in N, x \in S$. We have

$$v_{k+1}(x) = c(x) + \min_{A \subseteq N}\{r(A) + \mathrm{E}[v_k(y(x, A))]\},$$

and

$$v_{k+1}(x + \epsilon e^{(i)}) = c(x + \epsilon e^{(i)}) + \min_{A \subseteq N}\{r(A) + \mathrm{E}[v_k(y(x + \epsilon e^{(i)}, A))]\}.$$

We now compare $r(A) + \mathrm{E}[v_k(y(x, A))]$ and $r(A) + \mathrm{E}[v_k(y(x + \epsilon e^{(i)}, A))]$, for any fixed set $A \subseteq N$. We know $v_k(x) \geq v_k(x + \epsilon e^{(i)})$ by the induction hypothesis. By the definition of $y(x, A)$, we know $y^{(i)}(x, A) \leq y^{(i)}(x + \epsilon e^{(i)}, A)$ and $y^{(j)}(x, A) = y^{(j)}(x + \epsilon e^{(i)}, A),\ \forall j \neq i$. Hence we have

$$\mathrm{E}[v_k(y(x, A))] \geq \mathrm{E}[v_k(y(x + \epsilon e^{(i)}, A))], \quad \forall A \subseteq N.$$

This fact leads to

$$\min_{A \subseteq N}\{r(A) + \mathrm{E}[v_k(y(x, A))]\} \geq \min_{A \subseteq N}\{r(A) + \mathrm{E}[v_k(y(x + \epsilon e^{(i)}, A))]\}.$$

We know $c(x) \geq c(x + \epsilon e^{(i)})$ from the Assumption A. Hence, we have

$$v_{k+1}(x) \geq v_{k+1}(x + \epsilon e^{(i)}).$$

By induction, we have

$$v_k(x) \geq v_k(x + \epsilon e^{(i)}), \quad \forall k \geq 1, \epsilon \geq 0, i \in N, x \in S.$$

Hence, from Theorem 13, we have

$$\begin{aligned} h(x) - h(\mathbf{0}) &= \lim_{k \to \infty} [v_k(x) - v_k(\mathbf{0})] \\ &\geq \lim_{k \to \infty} [v_k(x + \epsilon e^{(i)}) - v_k(\mathbf{0})] \\ &= h(x + \epsilon e^{(i)}) - h(\mathbf{0}). \end{aligned}$$

The result $h(x) \geq h(x + \epsilon e^{(i)}),\ \forall \epsilon \geq 0, i \in N, x \in S$ follows since $h(\cdot)$ function inherits the structural properties of $v_k(\cdot)$ function. □

Remark: This theorem has an intuitive interpretation as follows. The bank starting from higher cash inventory in any ATM incurs less cost over the first k days, for all k.

Recall the definition of the optimal replenishment set $A(x)$ in state x as in Equation (3.5). We show $A(x)$ has the following property.

Theorem 18. *Under the Assumption A, $A(x)$ satisfies:*

$$i \in A(x + \epsilon e^{(i)}) \Rightarrow i \in A(x), \quad \forall \epsilon \geq 0, x \in S.$$

Proof. Note that $i \in A(x+\epsilon e^{(i)})$ gives

$$\min_{A\in\{B\subseteq N: i\in B\}}\{r(A)+\mathrm{E}h(y(x+\epsilon e^{(i)},A))\} \le \min_{A\in\{B\subseteq N: i\notin B\}}\{r(A)+\mathrm{E}h(y(x+\epsilon e^{(i)},A))\}.$$

By the definition of $y(x,A)$, we know $y^{(i)}(x,A) \le y^{(i)}(x+\epsilon e^{(i)},A)$ and $y^{(j)}(x,A) = y^{(j)}(x+\epsilon e^{(i)},A),\ \forall j \neq i$. By Theorem 17 we know $h(x)$ is a decreasing function in each component $x^{(i)}$ of the vector x, so $h(y(x+\epsilon e^{(i)},A)) \le h(y(x,A))$. Hence we have

$$\min_{A\in\{B\subseteq N: i\notin B\}}\{r(A)+\mathrm{E}h(y(x+\epsilon e^{(i)},A))\} \le \min_{A\in\{B\subseteq N: i\notin B\}}\{r(A)+\mathrm{E}h(y(x,A))\}.$$

By the definition of $y(x,A)$, we know

$$y^{(i)}(x+\epsilon e^{(i)},A) = (M_i - D^{(i)})^+ = y^{(i)}(x,A), \quad \forall i \in A.$$

Hence we have

$$\min_{A\in\{B\subseteq N: i\in B\}}\{r(A)+\mathrm{E}h(y(x+\epsilon e^{(i)},A))\} = \min_{A\in\{B\subseteq N: i\in B\}}\{r(A)+\mathrm{E}h(y(x,A))\}.$$

This yields that

$$\min_{A\in\{B\subseteq N: i\in B\}}\{r(A)+\mathrm{E}h(y(x,A))\} \le \min_{A\in\{B\subseteq N: i\notin B\}}\{r(A)+\mathrm{E}h(y(x,A))\} \Rightarrow i \in A(x).$$

$\square$

Remark: Intuitively, if it is optimal for the bank to replenish the ATM i with a certain amount of cash in it, then it is optimal for the bank to replenish this ATM when there is less cash in it, assuming that the cash inventory level at all other ATMs stay the same. This immediately raises a question: what can we say about the optimal replenishment set if the cash inventory levels in multiple ATMs change simultaneously? We give an interesting result about it next.

Recall $A(x)$ is the optimal replenishment set in the state of x. Define

$$Z(x) = \{(z^{(1)}, z^{(2)}, ..., z^{(n)}) \in S : z^{(i)} \leq x^{(i)},\ \forall i \in A(x) \text{ and } z^{(j)} = x^{(j)},\ \forall j \notin A(x)\}.$$

Theorem 19.

$$z \in Z(x) \Rightarrow A(x) \subseteq A(z), \quad \forall x \in S.$$

Proof. Fix an $x \in S$, suppose that

$$A(x) = \{i_1, i_2, ..., i_k\}.$$

Any $z \in Z(x)$ can be written as

$$z = x - \epsilon_{i_1} e^{(i_1)} - \epsilon_{i_2} e^{(i_2)} - ... - \epsilon_{i_k} e^{(i_k)},$$

where $\epsilon_{i_1}, \epsilon_{i_2}, ..., \epsilon_{i_k}$ are such that $0 \leq \epsilon_{i_t} \leq x^{(i_t)}$ for any $t = 1, 2, ..., k$. By Theorem 18, we know

$$i \in A(x) \Rightarrow i \in A(x - \epsilon_i e^{(i)}), \quad \text{for any } \epsilon_i \text{ such that } 0 \le \epsilon_i \le x^{(i)}.$$

We have

$$\begin{aligned} A(x) &\subseteq A(x - \epsilon_{i_1} e^{(i_1)}) \\ &\subseteq A(x - \epsilon_{i_1} e^{(i_1)} - \epsilon_{i_2} e^{(i_2)}) \\ &\subseteq \cdots \\ &\subseteq A(x - \epsilon_{i_1} e^{(i_1)} - \epsilon_{i_2} e^{(i_2)} - \ldots - \epsilon_{i_k} e^{(i_k)}) = A(z), \end{aligned}$$

$\square$

Remark: This theorem has an intuitive interpretation as follows. The optimal replenishment set gets larger while there is less amount of cash in the ATMs within the optimal replenishment set and same amount of cash in the ATMs outside the optimal replenishment set.

We now use numerical programs to verify the structural properties of the optimal policy in the two-ATM case. We use r_i to denote the replenishment cost at ATM i ($i = 1, 2$) and r_{12} to denote the cost of replenishing both ATMs. We use c_i to denote the stock-out costs at ATM i ($i = 1, 2$). We assume the stationary demand at ATM i follows a Poisson(λ_i) distribution. We use the following values of the parameters: $M_1 = M_2 = 50, r_1 = r_2 = 60, r_{12} = 100, \lambda_1 = \lambda_2 = 20, c_1 = c_2 = 76$. (This is a symmetric submodular case.) Figure

3.2 displays the optimal decision in each state.

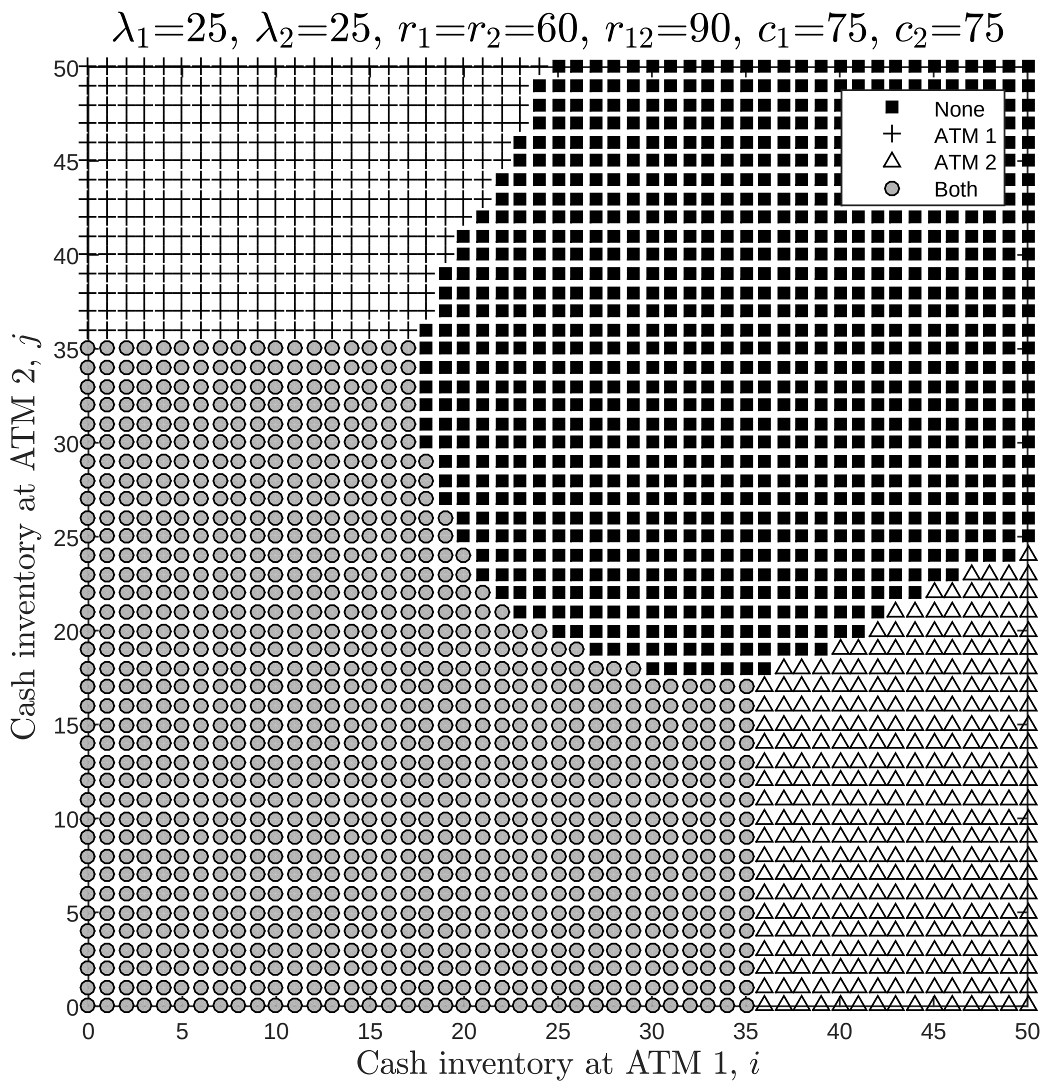

Figure 3.2: The state-dependent decisions under the optimal policy

Figure 3.2 illustrates that the conditions $z^{(j)} = x^{(j)}$, $\forall j \notin A(x)$ in the definition of set $Z(x, A)$ are critical. Any relaxation of these conditions can make the statement in Theorem 19 invalid. For example, we see from Figure 3.2 that

$$A(22, 45) = \{1\}, \quad A(20, 45) = \{1\}, \quad A(20, 35) = \emptyset.$$

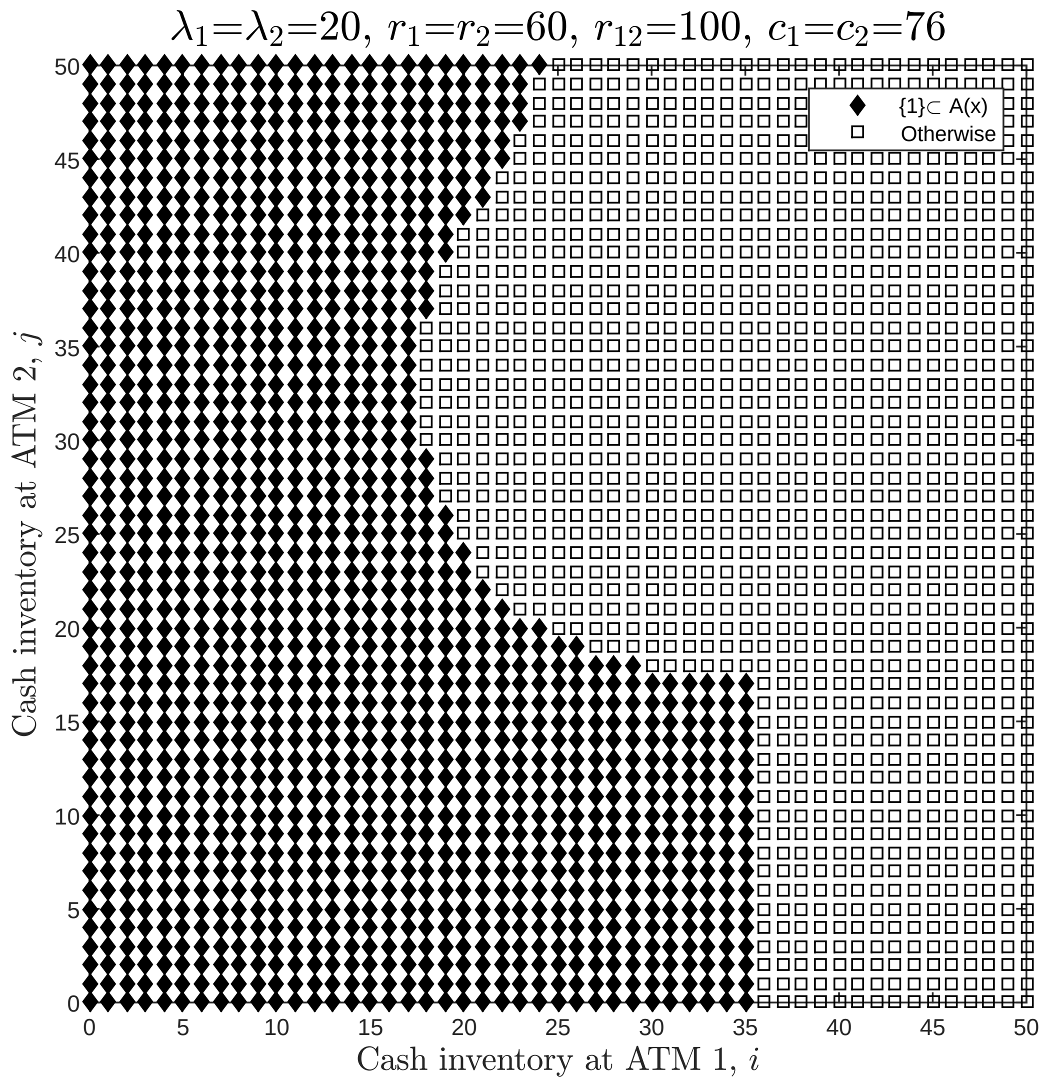

Figure 3.3: The decisions on ATM 1

Hence we see $A(22,45) \subseteq A(20,45)$, but $A(22,45) \nsubseteq A(20,35)$. Therefore, we see that decreasing the cash inventory level at ATM 2, which is not in the optimal replenishment set $A(22,45) = \{1\}$, changes the decision about ATM 1.

To visualize the structural pattern better, we plot the decisions made on ATM 1 and 2 in Figures 3.3 and 3.4 respectively. These figures also illustrate Theorem 17. For example, the optimal replenishment set include ATM 1 when the cash inventory in ATM 1 is lower

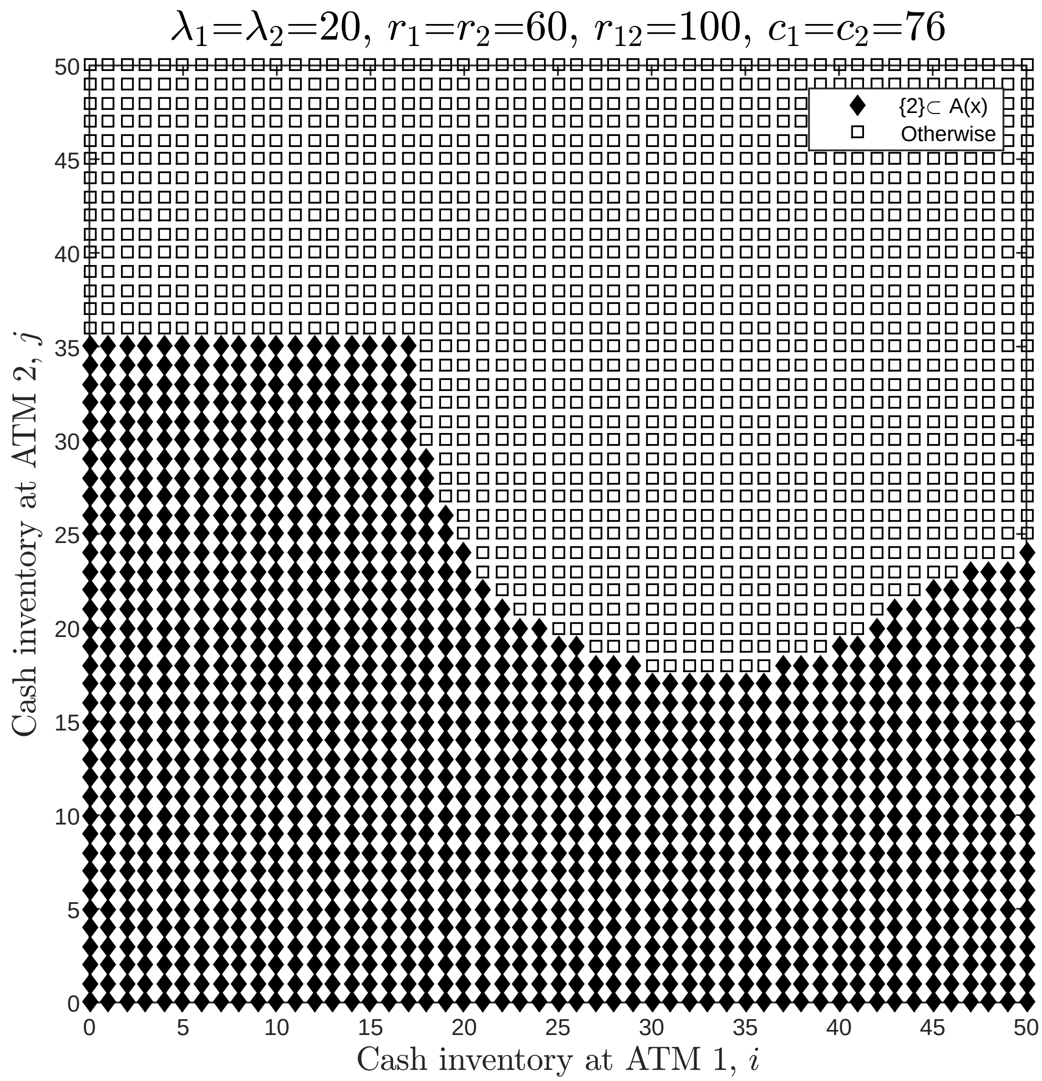

Figure 3.4: The decisions on ATM 2

than a certain threshold, as shown in Figure 3.3. We observe a similar pattern in Figure 3.4: the optimal replenishment set include ATM 2 when the cash inventory in ATM 2 is lower than a certain threshold. Those thresholds are functions of system parameters and the cash inventory in the other ATM.

3.7 Heuristic Policies

In this section, we focus on the realistic case where there are more than two ATMs. In general the optimal policies are impractical to compute due to the curse of dimensionality and combinatorially large number of decisions; and even harder to implement since they are not known in an analytical form. Hence we study two heuristic policies in this section. We focus on the case where the replenishment cost is non-additive, since the problem can be reduced to multiple single-ATM problems otherwise.

3.7.1 (s, M) Policy (SP)

As we have seen from Section 3.5, the (s, M) policy is optimal in the additive-replenishment-cost problem for some s. When the replenishment cost is not additive, the (s, M) policy is no longer optimal. However, it is a potentially easy-to-implement policy. We calculate the replenishment point $s^{(i)}$ for ATM i by minimizing the corresponding long-run average cost of operating ATM i only, where we use $r(\{i\})$ as the replenishment cost for ATM i. The (s, M) policy replenishes all ATMs where the cash inventory level is below the corresponding replenishment point. In mathematical terms, $A^{SP}(x)$ (where "SP" is short for "(s, M) policy"), the replenishment set in state x under this policy is given by:

$$A^{SP}(x) = \{i \in N : x^{(i)} < s^{(i)}\}.$$

It is easy to see that $A^{SP}(x)$ satisfies Theorems 18. Now define

$$Z^{SP}(x) = \{(z^{(1)}, z^{(2)}, ..., z^{(n)}) \in S : z^{(i)} \leq x^{(i)},$$
$$\forall i \in A^{SP}(x) \text{ and } z^{(j)} = x^{(j)},\ \forall j \notin A^{SP}(x)\}.$$

Then we have

Theorem 20.

$$z \in Z^{SP}(x) \Rightarrow A^{SP}(x) \subseteq A^{SP}(z), \quad \forall x \in S.$$

Proof. The proof is similar to the proof of Theorem 19. Hence we skip the proof. □

In other words, SP satisfies Theorem 19 and thus has the same structural properties as the optimal policy. We also know that, in the additive cost case, it is easy to find the s for which SP is in fact optimal. This is a strong argument in favor of considering it as a heuristic policy to consider.

3.7.2 Index Policy (IP)

We now develop an index policy parameterized by a scalar $p \in [0, 1]$. We begin with a randomized policy, which replenishes all ATMs with probability p or takes no action with probability $1 - p$, every day. In mathematical terms, $A^{RP}(x)$ (where "RP" is short for "randomized policy"), the replenishment set in state x under this policy is given by

$$A^{RP}(x) = \begin{cases} N, & \text{with probability } p \\ \emptyset, & \text{with probability } 1-p. \end{cases}$$

Clearly this is a state independent policy. We later elaborate on the choice of p in detail (see the later Equation 3.24).

The index policy is derived by applying one iteration of the policy improvement algorithm (see (Tijms, 2003)) to this randomized policy. We need the quantities in Lemma 7 below to find the index policy. Recall that $D_k^{(i)}$ is the amount of cash withdrawals that occur at ATM i on day k. We use $D^{(i)}$ to denote the generic random amount of daily cash withdrawals at ATM i.

Lemma 7. *Let $w^{(i)}(x^{(i)})$ denote the total stock-out cost of ATM i until the next replenishment starting from state $x^{(i)}$ under the randomized policy. Write $q = 1-p$. Then*

$$\mathrm{E}[w^{(i)}((x^{(i)} - D^{(i)})^+)] = c_i \phi^{(i)}(x^{(i)}), \quad \mathrm{E}[w^{(i)}((M_i - D^{(i)})^+)] = c_i \phi^{(i)}(M),$$

where

$$\phi^{(i)}(x^{(i)}) = \sum_{k=1}^{+\infty} q^{k-1} \cdot \mathrm{P}(D_1^{(i)} + \cdots + D_k^{(i)} \geq x^{(i)}). \tag{3.16}$$

Proof. We use two different methods to calculate $w^{(i)}(x^{(i)})$. The first method follows from the standard theory of dynamic programming. The stock-out cost incurred right before the beginning of today is $c_i\delta(x^{(i)})$. With probability p, the replenishment happens today and hence there is no more cost to be included in $w^{(i)}(x^{(i)})$. With probability q, the replenishment does not happen today. The state right before the beginning of tomorrow is $(x^{(i)} - D^{(i)})^+$, so the stock-out cost from then on until the first replenishment happens is $\mathrm{E}[w^{(i)}((x^{(i)} - D^{(i)})^+)]$. Thus

$$w^{(i)}(x^{(i)}) = c_i\delta(x^{(i)}) + q\mathrm{E}[w^{(i)}((x^{(i)} - D^{(i)})^+)], \tag{3.17}$$

where

$$\delta(x^{(i)}) = \begin{cases} 1, & x^{(i)} \le 0, \\ 0, & x^{(i)} > 0. \end{cases}$$

We now calculate $w^{(i)}(x^{(i)})$ in an alternative way. For any given $k \ge 0$, with probability $q^k p$, the replenishment happens k days later, the total stock-out cost until this replenishment is $c_i\delta(x^{(i)}) + c_i\mathrm{E}\delta((x^{(i)} - D_1^{(i)})^+) + \cdots + c_i\mathrm{E}\delta((x^{(i)} - D_1^{(i)} - \cdots - D_k^{(i)})^+)$. Hence we know

$$\begin{aligned} w^{(i)}(x^{(i)}) &= \sum_{k=0}^{+\infty} q^k p \left[c_i\delta(x^{(i)}) + \sum_{m=1}^{k} c_i\mathrm{E}\delta\left(x^{(i)} - \sum_{l=1}^{m} D_l^{(i)}\right)^+ \right] \\ &= c_i\delta(x^{(i)}) + \sum_{k=1}^{+\infty} q^k p \cdot \left(\sum_{m=1}^{k} c_i\mathrm{P}(D_1^{(i)} + \cdots + D_k^{(i)} \ge x^{(i)}) \right). \end{aligned}$$

We can rearrange this to get

$$w^{(i)}(x^{(i)}) = c_i\delta(x^{(i)}) + c_i q\phi^{(i)}(x^{(i)}). \tag{3.18}$$

Now we connect the two methods with each other. From Equations (3.17) and (3.18), we get

$$\mathrm{E}[w^{(i)}((x^{(i)} - D^{(i)})^{+})] = c_i \phi^{(i)}(x^{(i)}). \tag{3.19}$$

Setting $x^{(i)}$ to be M_i, we get

$$\mathrm{E}[w^{(i)}((M_i - D^{(i)})^{+})] = c_i \phi^{(i)}(M_i). \tag{3.20}$$

$\square$

We now apply one iteration of policy improvement algorithm to the randomized policy to derive an improved policy. Let $x = (x^{(1)}, ..., x^{(n)})$ and define $v^{RP}(x)$ to be the total cost (stock-out cost and replenishment cost) incurred until the next replenishment starting from state x, under the randomized policy. For a given $A \subseteq N$, we use π_A to denote the policy which replenishes the set A of ATMs today and follows the randomized policy RP starting from tomorrow. Let $v^{\pi_A}(x)$ be the total cost (stock-out cost and replenishment cost) incurred until the next replenishment occurs starting from state x, under the policy π_A. We then compute the quantity

$$v^{\pi_A}(x) - v^{RP}(x), \quad \forall A \subseteq N, x \in S.$$

This difference can be regarded as the relative cost incurred until the next replenishment when we choose A as our action today, instead of using randomized policy. This leads to an improved policy (which we call an index policy) as follows. At the beginning of each day, given the system state x, we replenish the ATMs in the set $A^{IP}(x)$ ("IP" is short for index policy), where

$$A^{IP}(x) = \arg\min_{A \subseteq N} \{v^{\pi_A}(x) - v^{RP}(x)\}. \tag{3.21}$$

We show how to determine $A^{IP}(x)$ in Theorem 21.

Theorem 21. *Let*

$$I(A, x) = r(A) + \sum_{i \in A} c_i(\phi^{(i)}(M_i) - \phi^{(i)}(x^{(i)})). \tag{3.22}$$

The $A^{IP}(x)$ in Equation (3.21) is given by:

$$A^{IP}(x) = \arg\min_{A \subseteq N} I(A, x). \tag{3.23}$$

Proof. We first compute $v^{\pi_A}(x)$. Given the state x, the stock-out cost is incurred and ATMs in the set A are replenished today with the replenishment cost $r(A)$ under policy π_A. For an ATM $i \in A$, the cash inventory at the beginning of tomorrow is $(M_i - D^{(i)})^+$. For an ATM $i \notin A$, the cash inventory at the beginning of tomorrow is $(x^{(i)} - D^{(i)})^+$. Hence we have

$$v^{\pi_A}(x) = \sum_{i=1}^{n} c_i \delta(x^{(i)}) + r(A) + \sum_{i \in A} \mathrm{E}[w^{(i)}((M_i - D^{(i)})^+)] + \sum_{i \notin A} \mathrm{E}[w^{(i)}((x^{(i)} - D^{(i)})^+)].$$

We now compute $v^{RP}(x)$. Given the state x, the stock-out cost is incurred. With probability p, all ATMs are replenished and with probability $q = 1 - p$ none of them is replenished.

Hence we have

$$v^{RP}(x) = \sum_{i=1}^{n} c_i \delta(x^{(i)}) + p\left(r(N) + \sum_{i\in N} \mathrm{E}[w^{(i)}((M_i - D^{(i)})^+)]\right)$$
$$+ q\left(\sum_{i\in N} \mathrm{E}[w^{(i)}((x^{(i)} - D^{(i)})^+)]\right).$$

By Lemma 7 and after some algebra we know

$$v^{\pi_A}(x) - v^{RP}(x) = r(A) - pr(N) + \sum_{i\in A} c_i[\phi^{(i)}(M_i) - \phi^{(i)}(x^{(i)})]$$
$$- p\sum_{i\in N} c_i[\phi^{(i)}(M_i) - \phi^{(i)}(x^{(i)})]$$

The only part of the right hand side above that involves A is

$$I(A, x) = r(A) + \sum_{i\in A} c_i(\phi^{(i)}(M_i) - \phi^{(i)}(x^{(i)})).$$

□

The function $I(A, x)$ is called the index of the set A in state x. The index policy chooses to replenish the ATMs in that set A in state x that has the lowest index. Hence we call this the index policy. It is known from the theory of MDPs that the performance of IP is superior to that of the RP.

The long-run average cost under the index policy is a function of p. To get the value of p which gives the lowest cost under the index policy, we can calculate the long-run average cost under the index policy by evaluating the values of p between 0 and 1 and then picking the p that yields the least cost. In the n-ATM problem for a large n, this method becomes intractable due to the computational load. Hence we develop a heuristic method of choosing p.

To define the choice of p formally, we introduce the following notation:

$$N^+ = \{i \in N : c_i > r(\{i\})\}$$
$$N^- = \{i \in N : c_i \leq r(\{i\})\}.$$

We choose

$$p = \begin{cases} 1, & \text{if } |N^+| \geq |N^-| \\ 0, & \text{if } |N^+| < |N^-|. \end{cases} \tag{3.24}$$

The above choice of p for the randomized policy is intuitively appealing: if there are more ATMs whose stock-out costs are larger than their stand-alone replenishment costs as compared to the ATMs whose stand-alone replenishment costs dominates the stock-out costs, we replenish all of them every day (that is, we choose $p = 1$); otherwise we choose never to replenish any ATMs. As we shall see, the index policy derived from improving this randomized policy seems to work very well. From now on we assume that the p is chosen according to Equation (3.24) when we use the index policy. Note that $\phi^{(i)}(x^{(i)})$ in Equation (3.16) becomes

$$\phi^{(i)}(x^{(i)}) = \begin{cases} f^{(i)}(x^{(i)}), & \text{if } p = 1 \\ \sum_{k=1}^{+\infty} \mathrm{P}(D_1^{(i)} + \cdots + D_k^{(i)} \geq x^{(i)}), & \text{if } p = 0. \end{cases}$$

The index $I(A, x)$ can now be derived using this $\phi^{(i)}(x^{(i)})$ in Equation (3.22).

Note that the minimization in Equation (3.23) is over all subsets of N. This is a combinatorially intractable problem in general. However, when the objective function in Equation (3.22) is submodular, a greedy algorithm can provide an efficient solution. The submodular structure of the objective function in a combinatorial optimization problem has been exploited by previous researchers to develop algorithms that can solve this problem efficiently. (Grötschel et al., 1981) proposed the first polynomial-time algorithm to minimize a submodular function, based on the ellipsoid method and the greedy algorithm. (Nemhauser et al., 1978) proved the celebrated result that the greedy algorithm gives a $(1 - \frac{1}{e})$-approximation for the optimal solution to maximizing a monotone submodular function subject to the cardinality constraint.

Recall the definition of a submodular function in Equation (3.1). The next theorem states one sufficient condition under which the objective function in our case, namely $I(A, x)$ in Equation (3.22), is submodular.

Theorem 22. *If $r(A)$ is a submodular function, then $I(A, x)$ in Equation* (3.22)*, is a submodular function of A, for all $x \in S$.*

Proof. It suffices to show that

$$I(A \cup \{i, j\}, x) - I(A \cup \{i\}, x) \leq I(A \cup \{j\}, x) - I(A, x), \quad \forall A.$$

Using Equation (3.22), it suffices to show that

$$r(A \cup \{i, j\}) - r(A \cup \{i\}) \leq r(A \cup \{j\}) - r(A), \quad \forall A.$$

This holds since $r(A)$ is a submodular function, which completes the proof. $\square$

Assuming that $I(A, x)$ is submodular, we propose a greedy algorithm to compute the solution $A^{IP}(x) = \arg\min_{A \subseteq N} I(A, x)$. Given the current state x, the system parameters $r(A)$, c_i, and the demand distributions, we can calculate $I(A, x)$ by Equation (3.22) for any $A \subseteq N$. Note that $I(\emptyset, x) = 0$ for any state x, so we can initialize the algorithm by setting $A = \emptyset$ and a temporary quantity $I_{\text{temp}} = 0$. We now calculate the index, denoted by $I(A \cup \{e\}, x)$, for all $e \notin A$. We compute e^* such that $I(A \cup \{e^*\}, x)$ is the smallest among all $I(A \cup \{e\}, x)$ for each $e \notin A$. Then we compare $I(A \cup \{e^*\}, x)$ with the temporary quantity I_{temp}. If the temporary quantity is no larger than $I(A \cup \{e^*\}, x)$, then it is less cost-effective to add any more elements to the set A. So we stop the algorithm. Otherwise, it is more cost-effective to add one more element, i.e. e^*, to the set A. Then we update the set A to A^*, set the value of I_{temp} to be $I(A \cup \{e^*\}, x)$, and repeat the above process. This procedure is formalized in Figure 3.5. Note that this greedy algorithm reduces the complexity of the problem in Equation (3.23) from $O(2^n)$ to $O(n^2)$.

Input: The current state x, the parameters $r(A), c_i$, and the demand distributions.
Output: Find the replenishment set A^*
Initialization: $A = \emptyset$ and $I_{\text{temp}} = 0$;
while $A \subseteq N$ **do**
    for $e \notin A$ **do**
        Choose $e^* = \arg\min_e \{I(A \cup \{e\}, x)\}$;
        if $I_{\text{temp}} \leq I(A \cup \{e^*\}, x)$ **then**
            stop;
        else
            Define $A^* \leftarrow A \cup \{e^*\}$;
            $I_{\text{temp}} = I(A^*, x)$;
            Update $A \leftarrow A^*$;
        end
    end
end

Figure 3.5: The greedy algorithm to find replenishment set A^*

It is not so obvious that the Index Policy also has the same structural properties as the optimal policy. Hence we state that as the next two theorems, and include their formal proof in the online supplement. Let $A^{IP}(x)$ be the replenishment set obtained by using the above greedy algorithm in state $x \in S$.

Theorem 23. *$A^{IP}(x)$ satisfies Theorem 18.*

Proof. Define $\hat{x} = x + \epsilon e^{(i)}$. Define

$$A^{IP}(\hat{x}) = \{\hat{a}_1, ..., \hat{a}_k, ..., \}$$

$$A^{IP}(x) = \{a_1, ..., a_k, ..., \}$$

where a_k is the element added to the set $A^{IP}(x)$ in the step k of the greedy algorithm and $\hat{a}_k$

is the counterpart of a_k when the system state is $\hat{x}$. We know

$$I(A,x) = r(A) + \sum_{i \in A} c_i(\phi^{(i)}(M_i) - \phi^{(i)}(x^{(i)})).$$

By this definition, we have

$$I(A,\hat{x}) = I(A, x + e^{(i)}) \begin{cases} \geq I(A,x), & \text{if } i \in A \\ = I(A,x), & \text{if } i \notin A \end{cases}$$

We now show $i \in A^{IP}(x)$ by considering two cases:

Case 1: If $\hat{a}_1 = i$, then we know

$$I(\{i\},\hat{x}) \leq I(\{j\},\hat{x}), \quad \forall j \neq i$$

Because

$$I(\{i\},x) \leq I(\{i\},\hat{x}), \quad I(\{j\},\hat{x}) = I(\{j\},x),$$

we have

$$I(\{i\},x) \leq I(\{j\},x), \quad \forall j \neq i$$

Hence we have $a_1 = i$ and $i \in A^{IP}(x)$.

Case 2: If $\hat{a}_1 = l \neq i$, then

$$I(\{l\}, \hat{x}) \leq I(\{j\}, \hat{x}), \quad \forall j \in N$$

For $j \neq i$, we know

$$I(\{l\}, x) = I(\{l\}, \hat{x}) \leq I(\{j\}, \hat{x}) = I(\{j\}, x).$$

For $j = i$, we have

$$I(\{l\}, \hat{x}) \leq I(\{i\}, \hat{x})$$

We know

$$I(\{l\}, \hat{x}) = I(\{l\}, x), I(\{i\}, x) \leq I(\{i\}, \hat{x})$$

Hence, we can not compare $I(\{l\}, x)$ and $I(\{i\}, x)$. So $a_1 = l = \hat{a}_1$ or $a_1 = i$. If $a_1 = i$, then we are done. Otherwise, $a_1 = \hat{a}_1$. Because $i \in A^{IP}(\hat{x})$ and $\hat{a}_1 \neq i$, we must have $\hat{a}_k = i$ for some $k \geq 2$. Define

$$\hat{A}_k = \{\hat{a}_1, ..., \hat{a}_k\},$$

$$A_k = \{a_1, ..., a_k\}.$$

We have

$$I(\hat{A}_{k-1} \cup \{i\}, \hat{x}) \leq I(\hat{A}_{k-1} \cup \{j\}, \hat{x}), \quad \forall j \in N \setminus \hat{A}_{k-1} \text{ and } j \neq i$$

and

$$I(\hat{A}_{k-1} \cup \{i\}, \hat{x}) \leq I(\hat{A}_{k-1}, \hat{x}).$$

If $i \in A_{k-1}$, then we are done. Otherwise, $i \notin A_{k-1}$ and we have $a_1 = \hat{a}_1, ..., a_{k-1} = \hat{a}_{k-1}$ by following the logic above. Hence $A_{k-1} = \hat{A}_{k-1}$ and we know

$$I(A_{k-1} \cup \{i\}, x) \leq I(\hat{A}_{k-1} \cup \{i\}, \hat{x}), \quad I(\hat{A}_{k-1} \cup \{j\}, \hat{x}) = I(A_{k-1} \cup \{j\}, x).$$

Hence,

$$I(A_{k-1} \cup \{i\}, x) \leq I(A_{k-1} \cup \{j\}, x), \quad \forall j \in N \setminus \hat{A}_{k-1} \text{ and } j \neq i.$$

Therefore $a_k = i$ and $i \in A^{IP}(x)$. $\square$

Now define

$$Z^{IP}(x) = \{(z^{(1)}, z^{(2)}, ..., z^{(n)}) \in S : z^{(i)} \leq x^{(i)},$$
$$\forall i \in A^{IP}(x) \text{ and } z^{(j)} = x^{(j)},\ \forall j \notin A^{IP}(x)\}.$$

Then we have

Theorem 24.

$$z \in Z^{IP}(x) \Rightarrow A^{IP}(x) \subseteq A^{IP}(z), \quad \forall x \in S.$$

Proof. The proof is similar to the proof of Theorem 19. Hence we skip the proof. □

In other words, IP satisfies Theorem 19 and thus has the same structural properties as the optimal policy. As in the SP case, this provides a strong justification to consider IP as another heuristic policy. Although both SP and IP have the same structural properties as the optimal policy, we show by numerical experiments that, in the submodular cost case, IP performs closer to the optimal policy compared to the SP. In the additive case, the performance of IP is close to that of SP, which we know is optimal.

3.8 Numerical Experiments

In this section, we compare the performance of the index policy (IP) and the (s, M) policy (SP) with that of the optimal policy (OP). We treat two cases separately: (1) the additive-cost case and (2) the submodular-cost case. In the additive-cost case, we know that SP is optimal and hence we compare IP with SP. For the submodular-cost case, we can compute the performance of OP, IP and SP numerically in two- and three-ATM cases. When there are four or more ATMs, computing the performance of OP, IP and SP, is computationally prohibitive due to the curse of dimensionality. Hence, for more than three ATMs we use simulation to compare IP to SP, but not to OP.

We have found that the total demand and demand heterogeneity within the ATMs have major impact on the performance of IP and SP compared to OP. Hence we introduce two measures to quantify the demand heterogeneity: the Gini coefficient, and the Coefficient of Variation (CoV).

Gini coefficient is a commonly used measure to quantify the wealth inequality. We can employ it to measure the heterogeneity of the demands. The Gini coefficient is defined as follows. We arrange the demand means $\lambda_1, ..., \lambda_n$ in a non-decreasing order and the ordered quantities are denoted by $\lambda_{(1)} < ... < \lambda_{(n)}$. The Gini coefficient is then given by:

$$\text{Gini} = \frac{n+1}{n} - \frac{2}{n} \cdot \frac{\sum_{i=1}^{n}(n+1-i)\lambda_{(i)}}{\sum_{i=1}^{n}\lambda_{(i)}}. \tag{3.25}$$

Gini coefficient is lower when the distribution of demand is more uniform. For example, if all the demand means are the same then the Gini coefficient is 0. The maximum Gini coefficient among n numbers is $\frac{n-1}{n}$ and occurs when $\lambda_{(1)} = \cdots = \lambda_{(n-1)} = 0$ and $\lambda_{(n)} > 0$.

To compute the CoV, we treat $(\lambda_1, \cdots, \lambda_n)$ as data, and compute its sample mean and sample standard deviation, and then compute CoV as the ratio (Sample Standard Deviation)/(Sample Mean). In the following, we fix the mean demand, vary the heterogeneity, and study its effects on the performance of various policies.

3.8.1 Comparison of IP, SP and OP in Two- and Three-ATM Case

We now compare the performance of IP and SP with that of OP in the two- and three-ATM case. We use $M_1 = M_2 = 50$ in the two-ATM case and $M_1 = M_2 = M_3 = 30$

in the three-ATM case. It takes about 30 seconds to run each case in the two-ATM setting, while it takes about 30 minutes each to run the three-ATM cases. (If we used $M = 50$ in the three-ATM case, it would take several hours for each case.) The daily cash demand follows the distribution of Poisson(λ_i) at ATM i (for $i = 1, \cdots, n$). The notation $(\lambda_1, \cdots, \lambda_n)$ is used to represent that ATM i has Poisson(λ_i) demand (for $i = 1, \cdots, n$). For example, in the two-ATM case, $(20, 25)$ means that ATM 1 has a Poisson demand with mean 20 and ATM 2 has a Poisson demand with mean 25.

We introduce two replenishment cost functions that are used through all the numerical studies. In general, the commonly used replenishment cost function in the joint replenishment or inventory routing literature is a fixed cost per order plus a common ordering cost. Thus, this replenishment cost function has the affine form. Specifically, we use the replenishment cost function:

$$r_{\text{affine}}(A) = \begin{cases} 30|A| + 300, & \text{if } A \neq \emptyset \\ 0, & \text{if } A = \emptyset. \end{cases}$$

The second replenishment cost function makes use of the known results about the length of the optimal traveling salesman tours. It has been shown, in Beardwood-Halton-Hammersley Theorem (see Theorem 2.4.1 in (Steele, 1997)), that the length of the optimal traveling salesman tours among n points which are uniformly distributed in the square of length 1 is asymptotically $O(\sqrt{n})$. Hence we use a square root form for the second replenishment cost function. Specifically, we use the replenishment cost function:

$$r_{\text{sqrt}}(A) = 60\sqrt{|A|}.$$

Note that both replenishment cost functions are submodular.

The values of stock-out cost estimates used in the two-ATM case (where the capacity of each ATM is 50) and three-ATM case (where the capacity of each ATM is 30) are given in Tables 3.1 and 3.2 respectively. Let c_{affine} and c_{sqrt} be the stock-out cost estimate under the replenishment cost function $r_{\text{affine}}(A)$ and $r_{\text{sqrt}}(A)$ respectively. They are chosen to ensure a service level of at least 99% for each ATM using the method of Section 3.4. We use them in Equation (3.2). Tables 3.1 and 3.2 also imply that we use $p = 1$ in the index policy.

λ	c_{affine}	c_{sqrt}
10	392	72
12.5	552	101
15	755	138
17.5	702	128
20	1794	327
22.5	2544	463
25	2305	420
27.5	1177	214
30	627	114
32.5	478	87
35	512	93

Table 3.1: Stock-out costs: $M = 50$

We use the value iteration algorithm to compute the long-run average cost g and Equation (3.9) as the stopping criterion. Our algorithm stops when the absolute error is less than $\epsilon = 0.01$. We have observed that the relative error is less than 0.01% of the optimal value when the algorithm stops. We use LRAC^{OP}, LRAC^{IP}, and LRAC^{SP} to denote the long-run

λ	c_{affine}	c_{sqrt}
6	565	103
9	1183	215
10	1231	224
12	2282	415
14	2554	465
15	1726	314
16	1734	316
18	957	174

Table 3.2: Stock-out costs: $M = 30$

average cost under OP, IP and SP respectively. We define

$$\mathrm{Gap}^{\mathrm{IP}} = \frac{\mathrm{LRAC}^{\mathrm{IP}} - \mathrm{LRAC}^{\mathrm{OP}}}{\mathrm{LRAC}^{\mathrm{OP}}}, \quad \mathrm{Gap}^{\mathrm{SP}} = \frac{\mathrm{LRAC}^{\mathrm{SP}} - \mathrm{LRAC}^{\mathrm{OP}}}{\mathrm{LRAC}^{\mathrm{OP}}}.$$

We include the results in Tables 3.3 and 3.4 for the case of the affine $r(\cdot)$. (We do not give the results for the square root $r(\cdot)$ because they are qualitatively similar to the affine case.) In both tables, the first column gives the vector of demand means. The next three columns display the mean of total demands faced by all ATMs λ_{sum} (which is the sum of demand means at all ATMs), the Gini coefficient, and the Coefficient of Variation. The next three columns present the long-run average cost g under OP, IP, and SP respectively. The last two columns give the gap between IP and OP and the gap between SP and OP. In all scenarios, IP performs close to OP (the gap is less than 5%) and better than SP. Note that IP performs closer to OP when the system faces a higher demand. We think this is the result of the fact that all the policies replenish more frequently when the demand is high, and hence there is less divergence among them. Also, at lower total demand, the performance of both the heuristic policies improves as the heterogeneity increases. At higher total demand, the performance of IP and SP is close to that of OP, but the effects of heterogeneity are less pronounced.

Demand	λ_{sum}	Gini	CoV	LRAC^{OP}	LRAC^{IP}	LRAC^{SP}	Gap^{IP}	Gap^{SP}
$(20, 20)$	40	0	0	249.8	261.0	308.7	4.5%	23.6%
$(22.5, 17.5)$	40	0.06	0.18	292.4	302.7	326.8	3.5%	11.8%
$(25, 15)$	40	0.13	0.35	332.9	334.0	343.9	0.3%	3.3%
$(27.5, 12.5)$	40	0.19	0.53	337.2	338.0	342.3	0.2%	1.5%
$(30, 10)$	40	0.25	0.71	336.7	337.2	339.9	0.1%	1.0%
$(22.5, 22.5)$	45	0	0	340.8	341.2	374.1	0.13%	9.8%
$(25, 20)$	45	0.06	0.16	345.2	345.7	362.7	0.14%	5.1%
$(27.5, 17.5)$	45	0.11	0.31	343.7	344.0	351.0	0.06%	2.1%
$(30, 15)$	45	0.17	0.47	343.0	343.4	349.7	0.12%	2.0%
$(32.5, 12.5)$	45	0.22	0.63	341.8	342.4	345.2	0.18%	1.0%
$(35, 10)$	45	0.28	0.79	343.2	343.5	345.2	0.10%	0.6%
$(25, 25)$	50	0	0	359.3	359.3	378.6	0.00%	5.4%
$(27.5, 22.5)$	50	0.05	0.14	358.2	358.2	380.1	0.00%	6.1%
$(30, 20)$	50	0.10	0.28	352.8	352.9	365.4	0.03%	3.6%
$(32.5, 17.5)$	50	0.15	0.42	347.5	347.6	352.5	0.04%	1.4%
$(35, 15)$	50	0.20	0.57	349.1	349.4	354.0	0.10%	1.4%

Table 3.3: Comparisons among OP, IP, and SP in 2-ATM affine-$r(A)$ case

Demand	λ_{sum}	Gini	CoV	LRAC^{OP}	LRAC^{IP}	LRAC^{SP}	Gap^{IP}	Gap^{SP}
$(12, 12, 12)$	36	0	0	353.67	359.72	409.38	1.71%	15.8%
$(9, 12, 15)$	36	0.11	0.25	367.78	368.95	398.79	0.32%	8.4%
$(6, 12, 18)$	36	0.22	0.50	371.22	372.02	389.84	0.22%	5.0%
$(14, 14, 14)$	42	0	0	390.29	390.29	432.58	0.00%	10.8%
$(12, 14, 16)$	42	0.06	0.14	389.05	389.05	423.16	0.00%	8.8%
$(10, 14, 18)$	42	0.13	0.29	384.37	384.37	407.14	0.00%	5.9%

Table 3.4: Comparisons among OP, IP, and SP in 3-ATM affine-$r(A)$ case

3.8.2 Comparison of OP and IP in Additive-Cost Multi-ATM Case

Note that OP is analytically intractable due to the curse of dimensionality and combinatorially large number of decisions, when there are more than three ATMs. However, there is one special case when the optimal policy is known in the multi-ATM setup: the additive-replenishment-cost case, when the optimal policy is of (s, M) type, as shown in Section 3.4. Although the performance of the OP can be computed numerically, it is not easy to compute the performance of IP in the muti-ATM additive-cost case. Hence, we design a simulation study below to compare the performance of OP and IP for the multi-ATM case when the replenishment cost is additive under the following parameters. This also helps us compare the performance of IP and OP on a sample-path basis.

There are $n = 30$ ATMs, so N is $\{1, 2, ..., 29, 30\}$, each with capacity $M = 50$. The cost for replenishment is 60, same for all ATMs. The stock-out costs are computed to guarantee the 99% of service level (the actual values of stock-out costs are included in the column c_{sqrt} of Table 3.1). The daily cash demand follows the distribution of Poisson(λ_i) at ATM i. The vector of demand means $(n_1 \times \lambda_1, n_2 \times \lambda_2, ..., n_k \times \lambda_k)$ is used to represent the case where n_j ATMs have Poisson(λ_j) demands for $j = 1, 2, ..., k$. For example, $(12 \times 20, 18 \times 25)$ indicates that 12 ATMs have a demand with mean 20 and 18 ATMs have a demand with mean 25.

We now explain the design of the simulation study. We use the common random numbers (CRN) technique to compare two stochastic systems. The CRN is a variance reduction technique, which works well when the performance outputs from two systems are positively correlated. We generate the daily random demands for all days in each of the $R = 100$ replications, with each replication simulating the system for $T = 2000$ days. We implement both OP and IP under the same demand stream. The long-run average cost is calculated by the sum of replenishment costs and stock-out costs in all days and all ATMs divided by the number of ATM-days (that is, 30*2000).

Table 3.5 displays the vector of demand means, the λ_{sum}, the Gini coefficient and the

Coefficient of Variation in the first four columns. The fifth column gives the mean of $\mathrm{Gap}^{\mathrm{IP}}$ from 100 replications. The last two columns give the mean of $\mathrm{LRAC}^{\mathrm{OP}}$ and $\mathrm{LRAC}^{\mathrm{IP}}$ from 100 replications respectively. As one can see from Table 3.5, the gap between IP and OP is less than 5% in all cases. IP performs closer to OP when the system faces a higher demand on average. For a fixed total demand the performance gap between IP and OP increases with the Gini coefficient. On the surface, this appears to be opposite of what we observed in the two and three ATM case. However, if we compare the IP with SP in those cases, the result is consistent. We shall see further results in the next subsection.

Demand	λ_{sum}	Gini	CoV	$\mathrm{Gap}^{\mathrm{IP}}$	$\mathrm{LRAC}^{\mathrm{OP}}$	$\mathrm{LRAC}^{\mathrm{IP}}$
$(4 \times 15, 24 \times 20, 2 \times 30)$	600	0.06	0.16	1.35%	1095.8	1110.6
$(9 \times 15, 18 \times 20, 3 \times 35)$	600	0.12	0.28	2.07%	1062.8	1084.8
$(15 \times 15, 10 \times 20, 5 \times 35)$	600	0.17	0.36	2.76%	1041.5	1070.2
$(10 \times 10, 10 \times 20, 10 \times 30)$	600	0.22	0.42	2.61%	1110.2	1139.2
$(10 \times 10, 10 \times 15, 10 \times 35)$	600	0.28	0.55	4.11%	1004.5	1045.8
$(15 \times 20, 15 \times 25)$	675	0.06	0.11	0.38%	1407.2	1412.5
$(3 \times 15, 19 \times 20, 6 \times 30, 2 \times 35)$	675	0.12	0.25	0.97%	1248.9	1261.0
$(8 \times 15, 13 \times 20, 4 \times 30, 5 \times 35)$	675	0.17	0.32	1.44%	1216.5	1234.1
$(2 \times 10, 15 \times 15, 5 \times 30, 8 \times 35)$	675	0.22	0.43	2.48%	1186.0	1215.4
$(15 \times 10, 15 \times 35)$	675	0.28	0.57	3.32%	1135.4	1173.1
(30×25)	750	0	0	0.02%	1717.9	1718.1
$(5 \times 20, 20 \times 25, 5 \times 30)$	750	0.06	0.12	0.15%	1626.2	1628.6
$(14 \times 20, 7 \times 25, 4 \times 30, 5 \times 35)$	750	0.12	0.23	0.37%	1455.5	1460.9
$(9 \times 15, 4 \times 20, 12 \times 30, 5 \times 35)$	750	0.17	0.32	1.17%	1389.3	1405.5
$(4 \times 10, 9 \times 15, 4 \times 30, 13 \times 35)$	750	0.22	0.42	1.90%	1311.8	1336.8

Table 3.5: Cost improvement versus Gini coefficient in 30-ATM additive-$r(A)$ case

3.8.3 Comparison of IP and SP in Submodular-Cost Multi-ATM Case

We now conduct a simulation study to compare the performance of IP and SP in the multi-ATM case when the replenishment cost is submodular, and not additive. Note that OP is analytically and numerically intractable in this case.

We use the same simulation design as in Section 3.8.2. We compute the long-run average cost as well as system-wide fraction of stock-out days per ATM per day. The system-wide fraction of stock-out days is calculated by the sum of stock-out-days experienced by all the ATMs divided by the total number of ATM-days. Let $\mathrm{LRAC}_\mathrm{k}^\mathrm{IP}$ and $\mathrm{FSOD}_\mathrm{k}^\mathrm{IP}$ be the long-run average cost and the system-wide fraction of stock-out days under IP in replication $k = 1, 2, ..., R$ respectively. Let $\mathrm{LRAC}_\mathrm{k}^\mathrm{SP}$ and $\mathrm{FSOD}_\mathrm{k}^\mathrm{SP}$ be the long-run average cost and the system-wide fraction of stock-out days under SP in replication $k = 1, 2, ..., R$ respectively. The performance improvement (from using IP over SP) in the long-run average cost in replication k is calculated by:

$$\mathrm{LRAC}_\mathrm{k}^\mathrm{improv} = \frac{\mathrm{LRAC}_\mathrm{k}^\mathrm{SP} - \mathrm{LRAC}_\mathrm{k}^\mathrm{IP}}{\mathrm{LRAC}_\mathrm{k}^\mathrm{SP}} \times 100\%$$

For example, suppose $\mathrm{LRAC}_\mathrm{k}^\mathrm{SP} = 100$ and $\mathrm{LRAC}_\mathrm{k}^\mathrm{IP} = 95$ in one replication. Then we say there is 5% cost improvement by implementing the index policy instead of (s, M) policy. This cost improvement is positive if the long-run average cost under SP is larger than that of IP, which happens if IP performs better. We then compute the mean of these R numbers. Similarly, the performance improvement in the system-wide fraction of stock-out days in replication k is calculated by:

$$\mathrm{FSOD}_\mathrm{k}^\mathrm{improv} = \frac{\mathrm{FSOD}_\mathrm{k}^\mathrm{SP} - \mathrm{FSOD}_\mathrm{k}^\mathrm{IP}}{\mathrm{FSOD}_\mathrm{k}^\mathrm{SP}} \times 100\%.$$

Figure 3.6 displays the comparison between the long-run average costs under the two policies in each of the 100 replications when there are 30 ATMs with the demand means $(5 \times 10, 5 \times 15, 5 \times 20, 5 \times 25, 5 \times 30, 5 \times 35)$ in the affine-$r(A)$ case. We observe that IP yields a lower cost compared to SP in every replication. Figure 3.7 displays the comparison between the system-wide fraction of stock-out days under two policies for each of the 100 replications. Note that the highest fraction of system-wide stock-out days is less than 1%, which is obtained from our required service level of 99%. However, the ATMs face much lower fraction of stock-out days by operating IP instead of SP. As we can see from these figures, IP performs better than SP in every replication. We have observed the same pattern in other parameter settings. We also observe the higher variance of both performance measures among different replications under SP, compared to those under IP.

Table 3.6 demonstrates the performance improvements corresponding to the assumed demand distributions under the affine replenishment cost function. The first four columns characterize the vector of demand means, the λ_{sum}, the Gini coefficient, and the Coefficient of Variation. The last two columns report the mean of $\text{LRAC}^{\text{improv}}$ from 100 simulation runs under the affine and square root replenishment cost respectively. As we can see from Table 3.6, the cost improvement mean decreases as the Gini coefficient increases, while the total demand is kept the same. Also, the gap between the performance of IP over that of SP is larger when the total demand is smaller. Both thses observations are consistent with those in Table 3.5. This pattern is observed under both affine and square root replenishment cost functions.

From all these experiments we conclude that the IP is a very useful policy to follow in general. A natural question is: how well will such a policy work in a real life situation

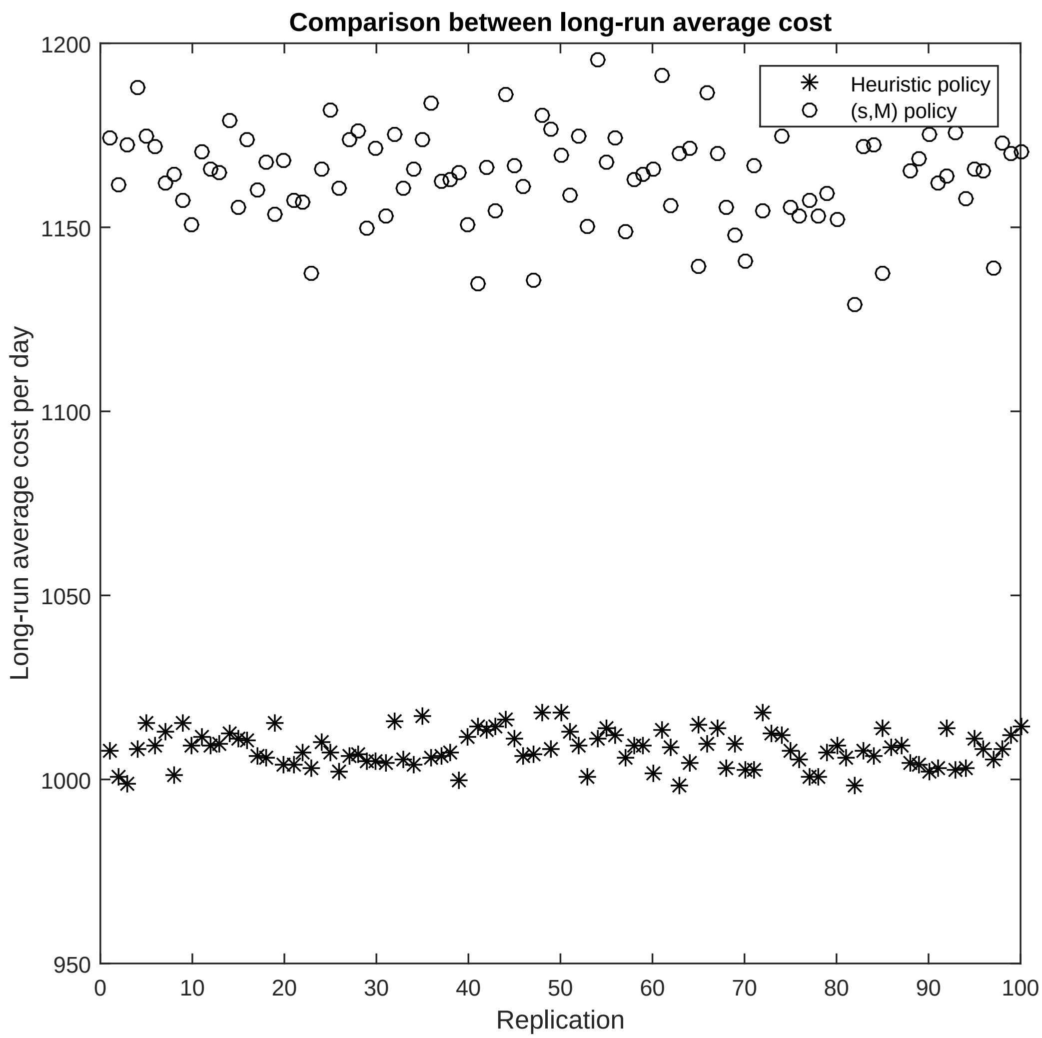

Figure 3.6: Long-run average cost

where the mathematical assumptions of the MDP model may not be completely satisfied? We consider precisely this question in the next section.

3.9 Real-world Applications

In this section we investigate whether the performance improvements observed in the numerical experiments hold in the real-world data. We have the historical data on the amounts of daily cash demands at each of the 139 ATMs belonging to a bank for a period of 20 months.

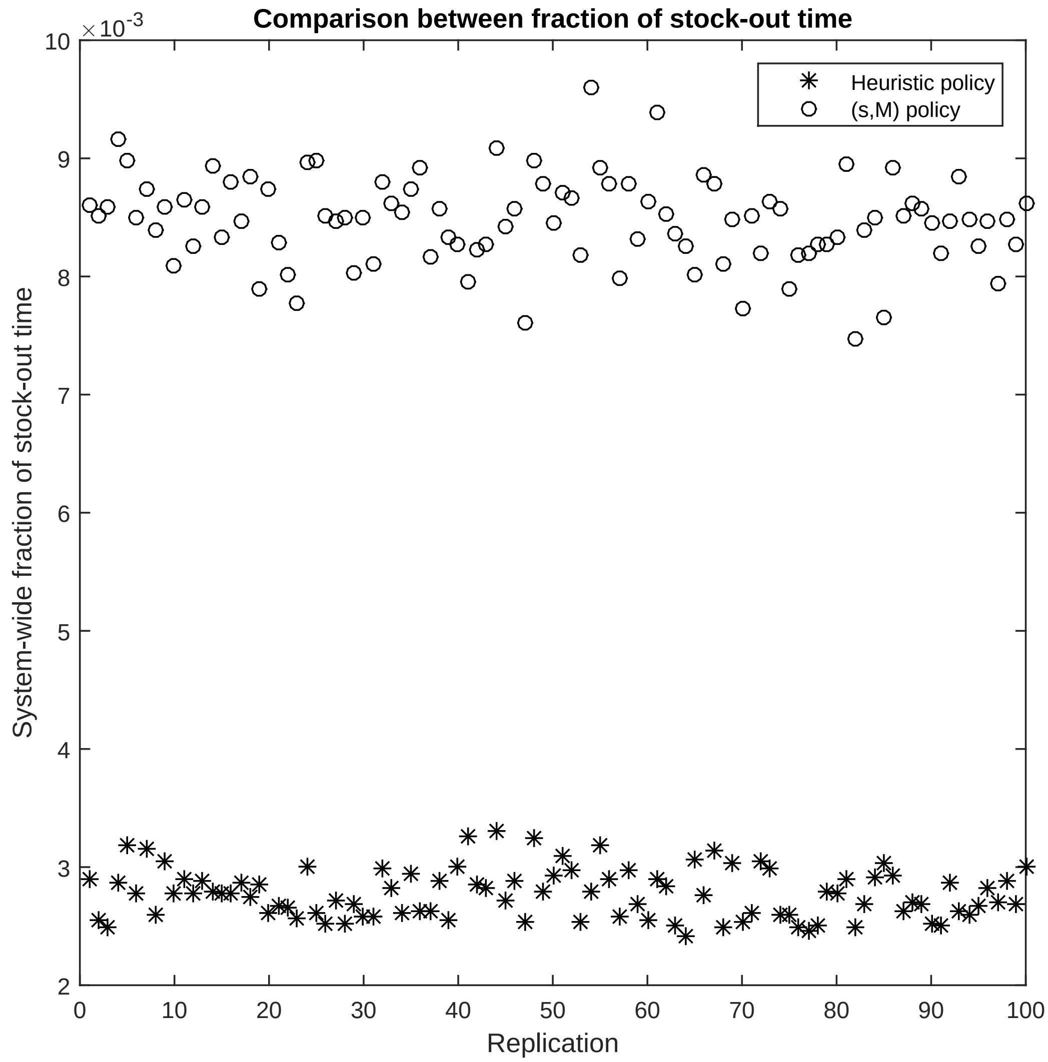

Figure 3.7: Fraction of stock-out days

In this section, we first conduct preliminary statistical analysis on the daily cash demands. We then explain different ways to estimate the demand distribution parameters needed for the models of Section 3.3. We finally implement the IP and SP based on the estimated distribution and the cost structures and compare their performances numerically using the real streams of demand data.

				LRAC$^{\text{improv}}$	
Vector of Demand Means	λ_{sum}	Gini	CoV	Affine	Square root
$(4 \times 15, 24 \times 20, 2 \times 30)$	600	0.06	0.16	21.57%	13.20%
$(9 \times 15, 18 \times 20, 3 \times 35)$	600	0.12	0.28	18.77%	11.15%
$(15 \times 15, 10 \times 20, 5 \times 35)$	600	0.17	0.36	15.56%	9.02%
$(10 \times 10, 10 \times 20, 10 \times 30)$	600	0.22	0.42	14.71%	8.63%
$(10 \times 10, 10 \times 15, 10 \times 35)$	600	0.28	0.55	6.61%	3.29%
$(15 \times 20, 15 \times 25)$	675	0.06	0.11	21.85%	14.71%
$(3 \times 15, 19 \times 20, 6 \times 30, 2 \times 35)$	675	0.12	0.25	18.06%	11.66%
$(8 \times 15, 13 \times 20, 4 \times 30, 5 \times 35)$	675	0.17	0.32	15.38%	9.68%
$(2 \times 10, 15 \times 15, 5 \times 30, 8 \times 35)$	675	0.22	0.43	8.84%	5.38%
$(15 \times 10, 15 \times 35)$	675	0.28	0.57	2.58%	1.05%
(30×25)	750	0	0	20.60%	14.67%
$(5 \times 20, 20 \times 25, 5 \times 30)$	750	0.06	0.12	19.29%	13.43%
$(14 \times 20, 7 \times 25, 4 \times 30, 5 \times 35)$	750	0.12	0.23	16.87%	11.41%
$(9 \times 15, 4 \times 20, 12 \times 30, 5 \times 35)$	750	0.17	0.32	11.15%	7.22%
$(4 \times 10, 9 \times 15, 4 \times 30, 13 \times 35)$	750	0.22	0.42	5.95%	3.65%

Table 3.6: Cost improvement versus Gini coefficient in 30-ATM case

3.9.1 Statistical Analysis

In this section, we conduct the statistical analysis of the historical daily cash demands. We first look at the daily cash demand over 20 months (From January 2010 to August 2011) for one ATM. We then observe evidence on the lack of seasonality of the daily cash demand. Finally we see that Gamma distribution is a good distribution fit for the daily cash demand.

Figure 3.8 displays the scaled daily cash demands versus the date. We do not observe a particular trend from Figure 3.8. We also use Dickey-Fuller test of a unit root in a time series to test whether the stochastic process is stationary. Consistent with our visual observation, the null hypothesis that there is a unit root (which means the stochastic process has a periodic component) can be rejected at a significance level of 0.001.

We have used autocorrelation function (ACF) and partial autocorrelation function (PACF)

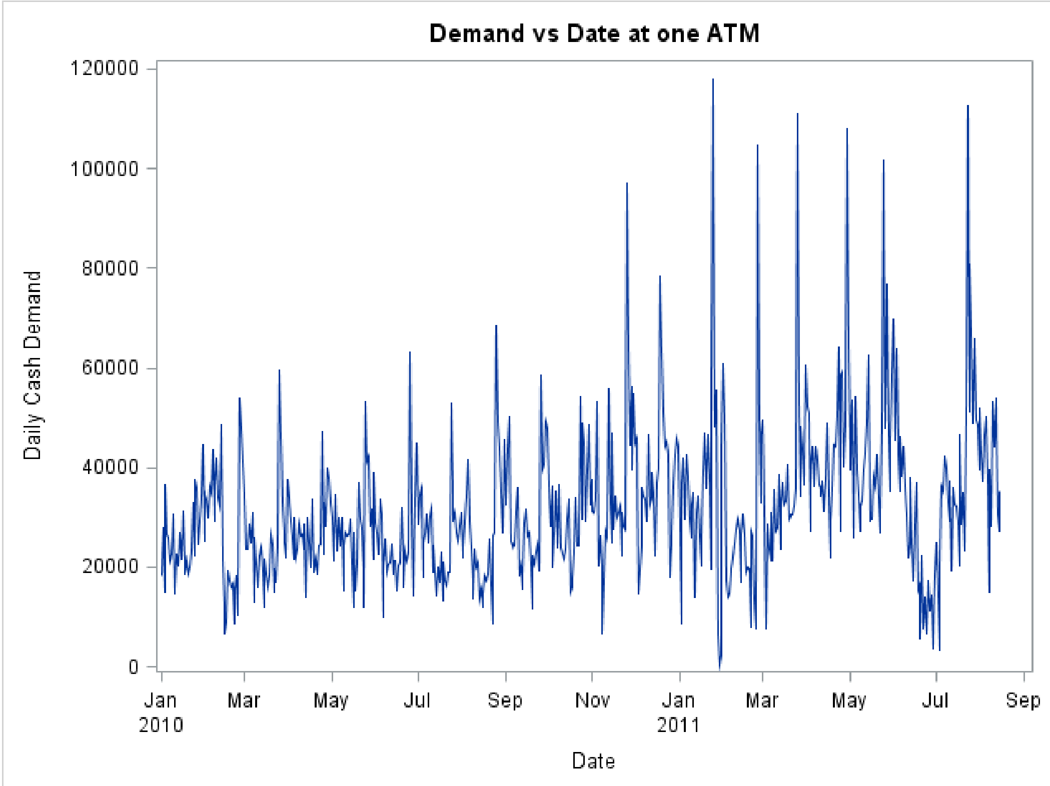

Figure 3.8: Demand vs Date

of the demand processes and found no seasonal effect. The lack of seasonality can also be observed from Figure 3.9. Each curve in Figure 3.9 represents one week. There are seven data points in each curve and each of them is the cash demand on that particular day of the week.

Since there is no specific weekly pattern, we conclude that the daily cash demand follows a certain distribution. Figure 3.10 displays the histogram and the fitted Gamma distribution. The shape of the histogram is close to the Gamma density distribution curve. We choose Gamma distribution due to the non-negativity of cash withdrawals and the right skewness observed in Figure 3.10. Several goodness-of-fit tests for Gamma distribution (Kolmogorov-Smirnov, Cramer-von Mises, and Anderson-Darling) are significant at 0.001 level. We de-

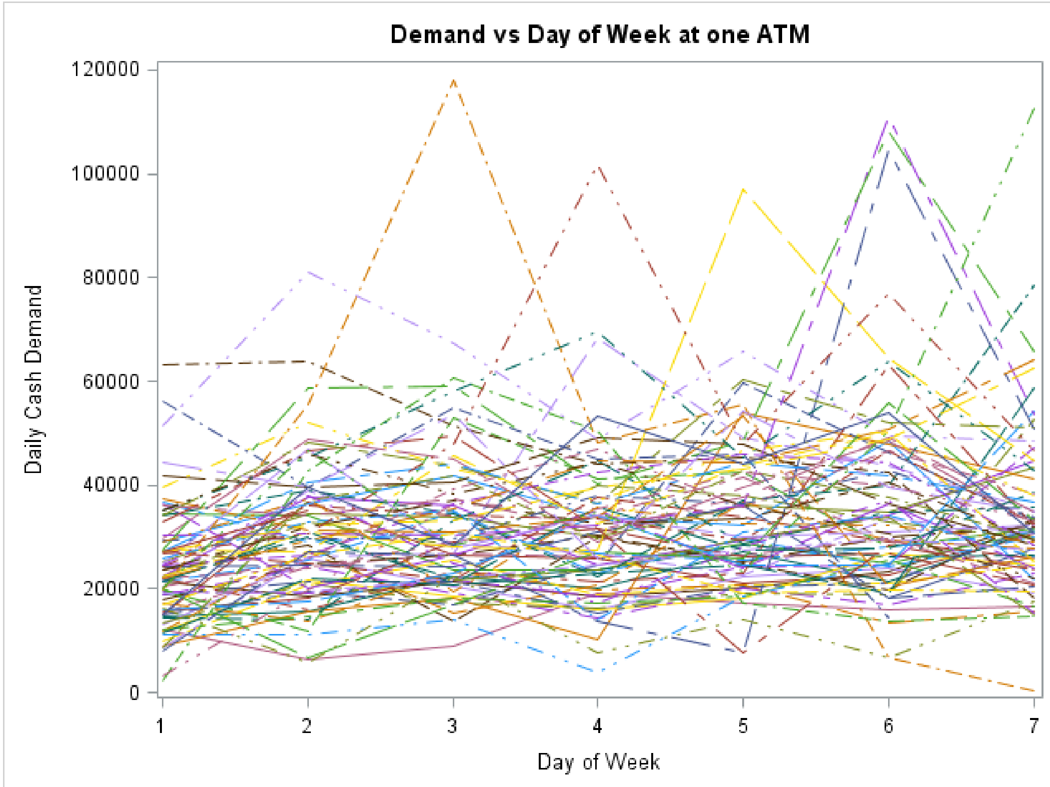

Figure 3.9: Demand vs Day of Week

scribe how to estimate the parameters of Gamma distribution in the next section.

We comment on the dependencies of the cash demands below. Note that the dataset indicates that the ATMs are not co-located within the same bank. Hence, no strong correlation is expected and observed from the data. (As one can imagine, the correlation between two ATMs within the same bank branch over 20 months is expected to be very high.) In our model, we assume that the demand for cash at one ATM is independent from the cash demand at another ATM. Intuitively, customers usually withdraw cash from the nearest and the most convenient one and they rarely go to another ATM 10 blocks away, unless it is empty. From the data, we do observe weak correlation between the demands of each ATM. However, we do not account for this factor in our model.

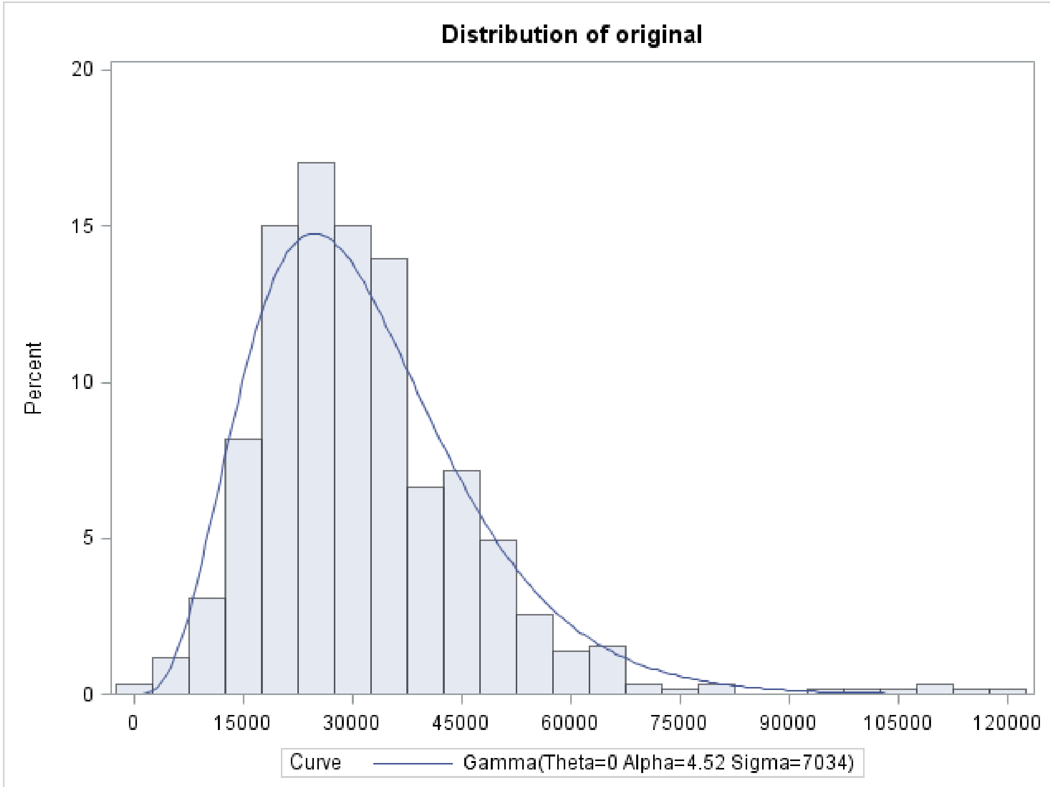

Figure 3.10: Demand Distribution

3.9.2 Parameter Estimation

We approximate the distribution of the daily demand by a Gamma(α, β) distribution. (The parameters are ATM-dependent.) To implement the IP and SP, we need the explicit form of the probability density function of the daily demand. Gamma density with shape parameter α and scale parameter β is given by

$$f(x|\alpha, \beta) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)} x^{\alpha-1} e^{-x/\beta}.$$

Its mean μ is $\alpha\beta$ and the standard deviation σ is $\sqrt{\alpha\beta^2}$. Suppose that we know the historical cash withdrawals $D_1, D_2, ..., D_k$ on day $1, 2, ..., k$ at the beginning of day $k+1$. We want to fit a Gamma distribution to these data. We now describe three different ways of parameter estimations in the Gamma distribution.

Maximum Likelihood Estimators

The Maximum Likelihood Estimator (MLE) of mean and standard deviation are given by

$$\hat{\mu}_{\mathrm{MLE}} = \frac{\sum_{i=1}^{k} D_i}{k},$$
$$\hat{\sigma}_{\mathrm{MLE}} = \sqrt{\frac{\sum_{i=1}^{k} (D_i - \hat{\mu}_{\mathrm{MLE}})^2}{k-1}}.$$

Then we see the MLE of α and β are given by

$$\hat{\alpha}_{\mathrm{MLE}} = \frac{\hat{\mu}_{\mathrm{MLE}}^2}{\hat{\sigma}_{\mathrm{MLE}}^2}, \quad \hat{\beta}_{\mathrm{MLE}} = \frac{\hat{\sigma}_{\mathrm{MLE}}^2}{\hat{\mu}_{\mathrm{MLE}}}.$$

Kernel Smoothing Estimators

The second estimator is borrowed from the kernel smoothing method in the non-parametric statistical models. The kernel smoothing estimator (KSE) is given by

$$\hat{\mu}_{\mathrm{KSE}} = \sum_{i=1}^{k} c_i D_i$$

where the weights $c_1, c_2, ..., c_k$ satisfy

$$c_i = C\phi\left(\frac{3(k-i)}{k}\right), \quad C = \frac{1}{\sum_{i=1}^{k}\phi\left(\frac{3(k-i)}{k}\right)}$$

and $\phi(\cdot)$ is the standard normal density. We calculate the second moment estimator similarly and compute the estimator of standard deviation by

$$\hat{\sigma}_{\mathrm{KSE}} = \sqrt{\sum_{i=1}^{k} c_i D_i^2 - \hat{\mu}_{\mathrm{KSE}}^2}.$$

Then the KSE of α and β are given by

$$\hat{\alpha}_{\mathrm{KSE}} = \frac{\hat{\mu}_{\mathrm{KSE}}^2}{\hat{\sigma}_{\mathrm{KSE}}^2}, \quad \hat{\beta}_{\mathrm{KSE}} = \frac{\hat{\sigma}_{\mathrm{KSE}}^2}{\hat{\mu}_{\mathrm{KSE}}}.$$

ARMA Estimators

In addition to MLE and kernel smoothing estimators, we also use the Autoregressive Moving Average (ARMA) method to estimate the mean and standard deviation. We fit an ARMA(1,1) to $\{D_1, ..., D_k\}$ and use this time series (TS) to predict the mean $\hat{\mu}_{\mathrm{TS}}$ and standard deviation $\hat{\sigma}_{\mathrm{TS}}$ of D_{k+1}. We then compute the shape and scale estimators by

$$\hat{\alpha}_{\mathrm{TS}} = \frac{\hat{\mu}_{\mathrm{TS}}^2}{\hat{\sigma}_{\mathrm{TS}}^2}, \quad \hat{\beta}_{\mathrm{TS}} = \frac{\hat{\sigma}_{\mathrm{TS}}^2}{\hat{\mu}_{\mathrm{TS}}}.$$

3.9.3 Numerical Study

We now conduct a numerical study to compare the performance of the IP and SP using the real-world dataset. Before doing so, we describe how and why we scale and discretize the daily cash demand. A common type of ATM has 4 cassettes and each cassette has 20 packets. Each packet contains 100 bills. Hence the ATM can contain at most $4 \times 20 \times 100 = 8000$ bills. In different geographical regions, people use different currencies and the most commonly used denomination varies currency by currency. For example, in the United States, ATMs usually use 20-dollar bills. Our dataset comes from the region where the most commonly used denomination is 50. Hence the capacity of each ATM is $400,000$ in the local currency. We divide the ATM's capacity and the daily cash demand by 5,000 and take the ceiling of these numbers to obtain the scaled capacity and scaled cash demand. We use the scaled numbers to conduct the statistical analysis and the numerical study. The reason of the scaling is to make the computation of the index more efficient.

The dataset has 592 days of daily cash withdrawals data from 139 ATMs. At the beginning of day k ($k \geq 57$), we estimate the daily demand distribution of D_k based on the cash withdrawal data $\{D_{n-56}, ..., D_{n-1}\}$ (8 weeks of data). We start from day 57 and implement the index policy and the (s, M) policy respectively. We use the demand data stream from the actual dataset as the demand for each day. The distance from the central vault to each ATM is also available. However, computing the optimal length of the traveling salesman problem (TSP) is an extremely hard problem. We refer the reader to (Applegate et al., 2011) for a monograph on TSP. It has been shown that the optimal length of the traveling salesman problem is *not* a submodular function; see (Herer, 1999). Let $T(S)$ denote the TSP optimal tour length for any subset of nodes S and let $K(S)$ be a submodular function. (Herer, 1999) has shown that there does not exist a constant $c > 1$ (independent of the graph) such that $T(S) \leq K(S) \leq cT(S)$ for all S. However, (Herer, 1999) proposed several heuristics to approximate the TSP optimal tour length by a submodular function and shows that the approximation error grows slowly with the number of nodes visited. We incorporate one of the approximations into our model. Mathematically, if we use $d_1, d_2, ..., d_n$ to denote the distance from the central vault to ATM $1, 2, ..., n$, then the replenishment cost for these ATMs is

$$2\left[\sqrt{\frac{n\pi}{2}} + \frac{3\pi}{2}\right] \cdot \max_{i=1,2,\ldots,n}\{d_i\}.$$

The stock-out costs are computed based on the distribution parameters obtained from the first 56 days using the replenishment cost and the service level requirement 98%. The stock-out costs are computed once and kept fixed. We now implement the index policy and the (s, M) policy using the three methods of parameter estimations discussed in Section 3.9.2. The results are included in Table 3.7.

	Average Cost		Service Level	
Parameter Estimation	Index Policy	(s, M) policy	Index Policy	(s, M) policy
MLE	1144.4	1816.3	99.8%	98.6%
Kernel Smoothing	1143.8	1843.7	99.8%	98.5%
ARMA(1,1)	1149.0	1792.6	99.8%	98.8%

Table 3.7: Performance comparison on the real-world dataset

As one can see from Table 3.7, the three estimation methods lead to similar results. The IP achieves a cost reduction of 35%∼40% over the SP while maintaining a better service level.

3.9.4 Recommendations

With the analysis presented above, we recommend that the bank should use the Maximum Likelihood Estimators to compute the parameter estimates for the demand distribution, since the MLE method is the simplest and it yields a relatively same result compared to the other two methods. Given the explicit form of the demand distribution, the bank can then compute the stock-out costs based on the replenishment costs and the demand distribution. Finally, the bank should implement our index policy to replenish the cash in the ATMs accordingly.

3.10 Summary and Future Work

We developed a model to design the ATMs replenishment schedule for the banks. The replenishment scheduling system presented the decisions on which ATMs to replenish dynamically, by taking into account the cash inventory levels of all ATMs, the demand distributions, the replenishment costs and the service level requirements. We constructed an MDP model and exploited the dynamic programming techniques to derive several structural properties of the optimal policy. We established the connections between the stock-out costs and the service level requirements. We proposed an index policy for the submodular-replenishment-cost case. We have shown by numerical study that the index policy performs close to the optimal policy in the two- and three-ATM submodular-cost cases as well as the mulit-ATM additive cost case. We illustrated by simulation that the IP outperforms the SP. We analyzed the real-world dataset of 139 ATMs with 20-month historical data on daily cash withdrawals. We showed by a numerical study that the index policy can reduce the cost by $35\% \sim 40\%$ with an improved service level.

There are several directions in which our current work can be extended. First, our analysis has assumed that the bank can replenish as many ATMs as the index policy recommends. However, in practice, there may be an upper bound on the number of replenishments possible. As mentioned in Section 3.1, this could be because the bank has to sort the available cash and fill the ATM cartridges. If b, the maximal number of ATMs that can be replenished is known, the index policy can be easily modified to incorporate this information. One can simply stop the algorithm in Figure 3.5 in Section 3.7.2 as soon as $|A| = b$. In the (s, M) policy one may need to rank the replenishment-eligible ATMs by a given criterion (say the level of cash left), and then replenish the top b of them. However, in reality the value of b may depend on the replenishment policy, especially when we take the cash left in the ATMs into consideration. This is a much more involved scenario and needs further research.

Another extension is to consider a different service quality indicator. In this work, we only consider the number of days when a stock-out occurs. We may use the amount of time when the ATM is empty as the service quality indicator in the future. Finally, we can generalize the current work to the situation where a replenishment can occur in the middle of the day and it takes a positive time (lead time) between the decision making and the actual replenishment. However, this generalization guarantees to be hard since the lost demand during the lead time requires further consideration.

REFERENCES

Applegate, D. L., Bixby, R. E., Chvatal, V., and Cook, W. J. (2011). *The Traveling Salesman Problem: A Computational Study: A Computational Study*. Princeton university press.

Arrow, K. J., Harris, T., and Marschak, J. (1951). Optimal inventory policy. *Econometrica: Journal of the Econometric Society*, pages 250–272.

Balasubramanian, H., Biehl, S., Dai, L., and Muriel, A. (2014). Dynamic allocation of same-day requests in multi-physician primary care practices in the presence of prescheduled appointments. *Health care management science*, 17(1):31–48.

Balintfy, J. L. (1964). On a basic class of multi-item inventory problems. *Management science*, 10(2):287–297.

Blok, H. and Spieksma, F. (2015). Structures of optimal policies in markov decision processes with unbounded jumps: the state of our art. *Technical report, Mathematisch Instituut, Universiteit Leiden*.

Burns, L. D., Hall, R. W., Blumenfeld, D. E., and Daganzo, C. F. (1985). Distribution strategies that minimize transportation and inventory costs. *Operations Research*, 33(3):469–490.

Cayirli, T. and Veral, E. (2003). Outpatient scheduling in health care: A review of literature. *Production and Operations Management*, 12(4):519–549.

Clark, A. J. and Scarf, H. (1960). Optimal policies for a multi-echelon inventory problem. *Management science*, 6(4):475–490.

Cordeau, J.-F., Laporte, G., Savelsbergh, M. W., and Vigo, D. (2007). Vehicle routing. In *Handbooks in Operations Research and Management Science: Transportation*, volume 14, pages 367–428.

Dekker, R. and Hordijk, A. (1992). Recurrence conditions for average and blackwell optimality in denumerable state markov decision chains. *Mathematics of Operations Research*, 17(2):271–289.

Dekker, R., Hordijk, A., and Spieksma, F. (1994). On the relation between recurrence and ergodicity properties in denumerable markov decision chains. *Mathematics of Operations Research*, 19(3):539–559.

Federgruen, A., Groenevelt, H., and Tijms, H. C. (1984). Coordinated replenishments in a multi-item inventory system with compound poisson demands. *Management Science*, 30(3):344–357.

Federgruen, A. and Simchi-Levi, D. (1995). Analysis of vehicle routing and inventory-routing problems. In *Handbooks in Operations Research and Management Science: Network Routing*, volume 8, pages 297–373.

Federgruen, A. and Zipkin, P. (1984). A combined vehicle routing and inventory allocation problem. *Operations Research*, 32(5):1019–1037.

Feldman, J., Liu, N., Topaloglu, H., and Ziya, S. (2014). Appointment scheduling under patient preference and no-show behavior. *Operations Research*, 62(4):794–811.

Goyal, S. K. and Satir, A. T. (1989). Joint replenishment inventory control: deterministic and stochastic models. *European journal of operational research*, 38(1):2–13.

Green, L. V. and Savin, S. (2008). Reducing delays for medical appointments: A queueing approach. *Operations Research*, 56(6):1526–1538.

Grötschel, M., Lovász, L., and Schrijver, A. (1981). The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197.

Gupta, D. and Denton, B. (2008). Appointment scheduling in health care: Challenges and opportunities. *IIE transactions*, 40(9):800–819.

Gupta, D. and Wang, L. (2008). Revenue management for a primary-care clinic in the presence of patient choice. *Operations Research*, 56(3):576–592.

Ha, A. Y. (1997). Optimal dynamic scheduling policy for a make-to-stock production system. *Operations Research*, 45(1):42–53.

Harris, F. W. (1990). How many parts to make at once. *Operations Research*, 38(6):947–950.

Hassin, R. and Mendel, S. (2008). Scheduling arrivals to queues: A single-server model with no-shows. *Management Science*, 54(3):565–572.

Herer, Y. and Roundy, R. (1997). Heuristics for a one-warehouse multiretailer distribution problem with performance bounds. *Operations Research*, 45(1):102–115.

Herer, Y. T. (1999). Submodularity and the traveling salesman problem. *European journal of operational research*, 114(3):489–508.

Iglehart, D. L. (1963). Optimality of (s, s) policies in the infinite horizon dynamic inventory problem. *Management science*, 9(2):259–267.

Ignall, E. (1969). Optimal continuous review policies for two product inventory systems with joint setup costs. *Management Science*, 15(5):278–283.

Kaandorp, G. C. and Koole, G. (2007). Optimal outpatient appointment scheduling. *Health Care Management Science*, 10(3):217–229.

Khouja, M. and Goyal, S. (2008). A review of the joint replenishment problem literature: 1989–2005. *European Journal of Operational Research*, 186(1):1–16.

Koole, G. (1998). Structural results for the control of queueing systems using event-based dynamic programming. *Queueing Systems*, 30(3-4):323–339.

Koole, G. (2007). Monotonicity in markov reward and decision chains: Theory and applications. In *Foundations and Trends in Stochastic Systems*, pages 1–76. Now Publishers Inc.

Kuiper, A., Kemper, B., and Mandjes, M. (2015). A computational approach to optimized appointment scheduling. *Queueing Systems*, 79(1):5–36.

Kulkarni, V. G. (2009). *Modeling and analysis of stochastic systems*. CRC Press.

LaGanga, L. R. and Lawrence, S. R. (2012). Appointment overbooking in health care clinics to improve patient service and clinic performance. *Production and Operations Management*, 21(5):874–888.

Liu, N., Ziya, S., and Kulkarni, V. G. (2010). Dynamic scheduling of outpatient appointments under patient no-shows and cancellations. *Manufacturing & Service Operations Management*, 12(2):347–364.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294.

Porteus, E. L. (1990). Stochastic inventory theory. In *Handbooks in Operations Research and Management Science: Stochastic Models*, volume 2, pages 605–652. Elsevier.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Scarf, H. (1959). The optimality of (s, s) policies in the dynamic inventory problem. In *Mathematical Methods in the Social Sciences 1959. Proceedings of the First Stanford Symposium*. Stanford University Press.

Spieksma, F. (1990). *Geometrically ergodic Markov Chains and the optimal control of queues*. PhD thesis, Leiden University.

Steele, J. M. (1997). *Probability theory and combinatorial optimization*, volume 69. Siam.

Talluri, K. and Van Ryzin, G. (2004). Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33.

Tijms, H. C. (2003). *A first course in stochastic models*. John Wiley and Sons.

Veinott Jr, A. F. and Wagner, H. M. (1965). Computing optimal (s, s) inventory policies. *Management Science*, 11(5):525–552.

Wang, W.-Y. and Gupta, D. (2011). Adaptive appointment systems with patient preferences. *Manufacturing & Service Operations Management*, 13(3):373–389.

Zacharias, C. and Pinedo, M. (2014). Appointment scheduling with no-shows and overbooking. *Production and Operations Management*, 23(5):788–801.

Zonderland, M. E., Boucherie, R. J., and Al Hanbali, A. (2015). Appointments in care pathways: the Geo^{x}/d/1 queue with slot reservations. *Queueing systems*, 79(1):37–51.