Stochastic Optimization Problems in the Service Industry with Customer Considerations

Atul Bhandari

Doctoral Dissertation

Submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in Operations Research
at the Tepper School of Business, Carnegie Mellon University.

September 2006

Dissertation Committee:

Alan Scheller-Wolf (Chair), Tepper School of Business, CMU

Gerard Cornuéjols, Tepper School of Business, CMU

Mor Harchol-Balter, School of Computer Science, CMU

Nicola Secomandi, Tepper School of Business, CMU

Geert-Jan van Houtum, Department of Technology Management, Technische Universiteit Eindhoven

Tepper School of Business, Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This dissertation studies the effect of customer considerations in stochastic optimization problems in the service industry. The first part looks at problems in the service industry geared towards providing efficient service, i.e., minimizing the total cost of providing service while ensuring that certain service level targets are met. In particular, we study applications in the call center industry, repair facilities, and spare parts industry. We develop mathematical programming models for these problems to determine the optimal non-randomized policy, i.e., a policy not involving any probabilistic mixing. The service level targets in these problems can be naturally modeled as constraints, which can be incorporated into a mathematical program. Despite this fact, to the best of our knowledge, this dissertation is the first study to use such a formulation to find an optimal non-randomized policy. The second part of this dissertation considers a revenue-management setting in the service industry but, different from the existing literature, the seller has to negotiate the price with each buyer. This is a typical situation in B2B applications such as telecommunications, cargo shipping, etc; in such settings the seller negotiates a different sale price with different buyers. We also consider a model in which the seller ensures that the average buyer is no worse-off by bargaining than under posted-pricing, via a constraint. We characterize how the seller and the average buyer are effected. We obtain several additional insights through numerical experiments.

To my family

Acknowledgements

I am extremely fortunate to have Alan Scheller-Wolf as my PhD advisor who taught me not only to be a researcher but also to be an academic. Without his guidance, encouragement, support, insights, and patience over all these years, this thesis would not have been possible. I greatly appreciate the generous amount of time and energy he spent on our discussions as well as on helping me develop my editorial skills. Thank you also for teaching me the importance of ethics in an academic setting.

The development of this thesis has greatly benefited from the excellent courses taught at Carnegie Mellon University (CMU) and its stimulating research environment that encourages cross-disciplinary work. This is in large part due to the exceptional faculty at the various schools of CMU. I have had the good fortune to work with and receive instruction and suggestions from several members of the faculty: Alan Scheller-Wolf, Egon Balas, Gerard Cornuéjols, Lisa Fleischer, Mor Harchol-Balter, John Hooker, Javier Peña, R. Ravi and Nicola Secomandi. I would also like to thank Geert-Jan van Houtum, a faculty member at the Technische Universiteit Eindhoven, for showing keen interest in my research.

A special shout goes out to Vincent Conitzer, Kevin Howard, Anand Iyer and David Maiwand. The memories of all our adventures in Pittsburgh will always be special and you were like family away from family. I am thankful to Amitabh Sinha and Jochen Könemann, my office mates when I first arrived at CMU and great friends ever since, for their support, mentoring, and free counseling. I am also thankful to Miroslav Karamanov and Tallys Yunes for their wonderful friendship and their infinite patience in helping me with UNIX and CPLEX commands and tricks. I would also like to thank Nihat Altinas, Kent Andersen, Ismail Civelek, Borga Deniz, Paul Enders, Vineet Goyal, Espen Henriksen, Latife Genc Kaya, Bram Kranenburg, Yanjun Li, Viswanath Nagarajan, Takayuki Osogami, Anureet Saxena, Mohit Singh, Erkut Sonmez, Jiong Sun, Xiaofang Wang, Xiaofeng Wang, Sunil Wattal, Adam Wierman, Chen Xiang and Hakan Yildiz for being wonderful friends and colleagues. Jackie Cavendish and Lawrence Rapp have been of immense help in dealing with the administrative aspects of the PhD program.

I am thankful to my grandparents for teaching me important values in life. I would like to thank my role model and brother, Amit, for everything he has done for me and for introducing me to the field of Operations Research. I thank him for purposely losing our very first chess match. I hope we become co-authors someday and become known as the "brothers in ORMS"! I am thankful to Nicole Bhandari for our discussions on dissertation writing and applying for faculty positions. I would like to thank my parents for their love and support and their belief in my abilities. It is impossible for me to list everything that you have done for me. I love you both. This thesis is as much your dream as it is mine and we finally did it.

I have to mention my two "Bubbies" (labrador retrievers): Bella and Danni. Thank you for your unconditional love and loyalty and for showing me how much pure joy there is in a simple game of fetch. Finally, I am grateful to my beautiful wife Kristen for her love, and for making everything in my life so much more meaningful. I could not have done it without you. Thank you for your support throughout the ups and downs of my PhD, and for your unwavering encouragement during the arduous task of writing this thesis.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This dissertation studies the effect of customer considerations on stochastic optimization problems in the service industry in two parts. The first part looks at problems in the service industry geared towards providing efficient service, i.e., minimizing the total cost of providing service while ensuring that certain service level targets are met. We develop mathematical programming models for these problems to determine the optimal non-randomized policy, i.e., a policy not involving any probabilistic mixing. The service level targets in these problems can be naturally modeled as constraints, which can be incorporated into a mathematical program. Despite this fact, this dissertation is the first study (to the best of our knowledge) to use such a formulation to find an optimal non-randomized policy. We discuss the different techniques we studied to solve these problems along with their potential strengths and weaknesses in chapter 2. In particular, we study applications in the call center industry, repair facilities, and spare parts industry (chapters 3 and 4). The second part considers a revenue-management setting in the service industry in which a seller has a finite amount of inventory to sell in a finite number of time periods. Different from the existing posted-price literature, the seller has to negotiate the price with each buyer. This problem has applications in business-to-business settings such as telecommunications and cargo shipping; in settings such as these, the seller typically negotiates a different sale price with different buyers. To do so we model the as solving solves a dynamic program which incorporates the Nash (1950) bargaining solution within the program. The optimal value functions of the seller from the bargaining model are shown to be always greater than those from the posted-price model, but the average buyer may be worse-off. A bargaining model which results in a Pareto improvement with respect to the seller's and the buyer's value functions relative to the posted-price model is also considered. Several additional insights are obtained through numerical experiments. Chapters 5 and 6 are dedicated to this topic. Below, a brief introduction of each chapter of this dissertation is provided. A more detailed introduction is also provided at the start of each chapter.

1.1 Chapter 2: Introduction to CMDPs

We introduce the basic theory of Markov chains, balance equations, limiting probability values and Markov decision processes (MDPs). A constrained Markov decision process (CMDP) is a MDP with probabilistic constraints. Each of the applications we study in the first part of this thesis can be modeled as a constrained Markov decision process (under suitable assumptions for the input parameters). However, until now the only exact solution method to obtain the optimal non-randomized policy for such problems is enumeration, which can be prohibitively expensive. We present four different mathematical programming formulations to obtain the optimal non-randomized policy for such problems in chapter 2: (i) the MBEDC algorithm, (ii) the BEDP algorithm, (iii) the BIDP algorithm, and (iv) the BIP algorithm. We then perform experiments to determine the efficiency of the BEDP, the BIDP and the BIP formulations. We conclude by discussing their relative strengths and weaknesses of each. Discussion of the MBEDC algorithm is presented in the following chapter, since it is the most efficient algorithm. If the reader is already familiar with CMDPs, this chapter can be skipped.

1.2 Chapter 3: The CDOS Problem

Call center managers are facing increasing pressure to reduce costs while maintaining acceptable service quality. Consequently, they often face constrained stochastic optimization problems, minimizing cost subject to service level constraints. Complicating this problem is the fact that customer arrival rates to call centers are often time-varying. Thus, to satisfy their service goals in a cost-effective manner, call centers may employ *permanent operators* who always provide service, and *temporary operators* who provide service only when the call center is busy, i.e., when the number of customers in system increases beyond a *threshold level*. This provides flexibility to dynamically adjust the number of operators providing service in response to the time-varying arrival rate. (Another way to cope with time varying arrival rate is to alter the staffing levels without temporary operators by breaking the day into different blocks or periods. This must be proactive, not reactive. In addition, with such fixed schedules temporary operators can still be utilized for each block.)

We define the Constrained Dynamic Operator Staffing (CDOS) problem as follows: determining the number of permanent and temporary operators, and the threshold value(s) that minimize time-average hiring and opportunity costs subject to service level constraints. We model the CDOS problem as a constrained Markov decision process (CMDP) and seek the optimal *non-randomized* policy. The only exact method in the literature to obtain the optimal non-randomized policy for a constrained MDP is enumeration, which is often computationally prohibitive. We provide a novel *exact* and *efficient* solution method, the MBEDC algorithm, yielding a Mixed Integer Program formulation; the computation times of this algorithm for sample problems are lower than enumeration by up to a factor of 200, and by a factor of 10 on average. Using our algorithm, we quickly solve diverse instances of the CDOS problem, generating managerial insights into the effects of temporary operators and different types of service level constraints.

1.3 Chapter 4: Other Applications of CMDPs

We study two other applications of stochastic optimization problems with service level constraints from the literature. The first has applications in call centers, repair facilities, etc., and studies the benefits of using a Beneficiary-Donor model in a multi-server system that consists of two servers and two queues: One of the servers (the Donor) may be asked to work on the jobs of the other server (the Beneficiary) but not vice versa. The objective is to minimize the weighted sum of the mean response times at the two servers by determining when the donor server should work on the beneficiary server's jobs and on its own jobs. Multiple threshold-based policies have been proposed and analyzed for this problem in the literature by different researchers, with the number of thresholds ranging from one to infinity. However, there is no work that answers the question of how many thresholds are actually required. Obviously, having more thresholds results in a better objective but also adds to the implementation complexity. On the other hand, if a reasonable objective value (within $p\%$ of the optimal) can be obtained by using only one or two thresholds, such a policy may be more practical. We are the first to provide a framework for comparing any threshold based policy to the optimal policy with respect to some given objective function. In particular, we compare a three-threshold policy - ADT (Osogami et al., 2005) - with the optimal policy with respect to the weighted sum of the mean response time metric. Ours is the first study to test the claim that the ADT policy performs close-to-optimal with respect to this metric. The general optimal policy is the solution to a Markov decision process and hence can be obtained by different techniques: policy iteration, value iteration, and linear programming. The restrictions to the class of ADT policies results in a constrained Markov decision process for which the optimal threshold values can be obtained by using the MBEDC algorithm developed in this dissertation. The framework used to compare these policies can be extended to compare any threshold-based policy with the optimal policy.

The second application is a multi-item, two location, spare parts problem with lateral transshipments, with the objective of finding the optimal non-randomized policy so as to minimize total inventory costs subject to waiting time constraints at each location. Such a problem is relevant for suppliers of complex machines in equipment-intensive industries such as airlines and manufacturing plants. A random failure of even a single part can cause the whole system to fail. As downtime can be very costly, the supplier typically maintains spare parts inventory at a warehouse near its clients' locations. At the same time, maintaining an excessive number of spare parts should also be avoided as most of these items are expensive. Hence, the suppliers are confronted with the difficult task of maintaining high system availability, while at the same time limiting the spare parts inventories.

If a demanded spare part is not available at the local warehouse, then the spare part is either shipped from a second local warehouse resulting in a lateral transshipment, or from the central warehouse resulting in an emergency shipment. While there exists a greedy heuristic to solve this problem, there is no exact solution method other than enumeration. This problem can be modeled as a constrained Markov decision process and we solve it exactly using the MBEDC algorithm developed in this dissertation.

1.4 Chapter 5: Revenue Management with Nash Bargaining

We consider the classical revenue-management setting of a firm selling a finite inventory of a single product within a finite number of periods. In each period there is a positive probability of a single buyer arrival. The buyer's reservation value for one unit of product is uncertain, but is drawn from a distribution known to the seller. Different from the existing literature, the seller *negotiates* the price with each buyer, a typical situation in business-to-business settings. We use the Nash bargaining solution to model the outcome of each negotiation and study the seller's optimal expected revenue.

We propose a model in which the seller dynamically optimizes its reservation value, establish structural properties of this model, and show that the average expected revenue benefit from adopting an optimal dynamic, rather than myopic, bargaining policy is 6-8%. A myopic policy refers to the seller maximizing its expected revenue for the current period without taking into account the effect on future revenue. We also show that the optimal value functions of our model are always higher than those obtained from the traditional posted-price model. We analyze this result by decomposing the difference between these value functions into a quantity effect and a price effect. Finally, we propose a hybrid model that provides a Pareto improvement over the posted-price model for both parties. In this case, the seller is only slightly worse-off while the buyers gain significantly in terms of expected surplus compared to the dynamic optimization bargaining model.

1.5 Chapter 6: Additional Experiments for Chapter 5

We present additional experiments for our bargaining model (from chapter 5) in this chapter, including a discussion of the concept of ex-post efficiency of our bargaining model. First, we extend the numerical examples from sections 5.2.2 and 5.3.2 to the cases when the buyer's reservation value is drawn from exponential and triangular distributions, obtaining similar results. While section 5.3.1 explains the improved performance of the optimal value functions of the bargaining model over the posted-price model, we show in this chapter that the immediate (or current) expected reward is also higher in the former model. We explain this result by studying the two factors of the immediate expected reward: (i) change in the probability of a sale agreement and (ii) change in the expected sale price conditional on a sale agreement. We show numerically that both these factors are positive for the bargaining model with respect to the posted-price model. Finally, we show that the bargaining model is not ex-post efficient. A bargaining outcome is ex-post efficient if and only if after all the information is revealed, the players' payoff associated with the bargaining outcome are Pareto-efficient. We consider an ex-post efficient bargaining model and show that it results in the highest benefit for the average buyer but significantly reduces the seller's expected revenue as compared to the bargaining model.

1.6 Chapter 7: Conclusions

Finally, we present conclusions from each of the chapters and discuss plans for future work in this chapter. We believe that some of these plans require only minor modifications of existing techniques (outlined here and elsewhere), whereas others are more long-term goals.

Chapter 2

Introduction to Constrained Markov Decision Processes

In this chapter, we discuss all formulations (solution techniques) that we studied to obtain the optimal non-randomized policy for any constrained Markov decision process. We present and discuss four formulations, two based on Disjunctive Programming (Balas, 1998), one based on Integer Programming (Nemhauser and Wolsey, 1988), and our most efficient algorithm, the MBEDC, based on Integer Programming. We defer the detailed discussion on the MBEDC algorithm to Chapter 3 and only introduce it in this chapter. Before discussing our algorithms, we introduce Markov Chains, Balance Equations, Markov Decision Processes, Constrained Markov Decision Processes, and present relevant notation. If the reader is already familiar with the concept of constrained Markov decision processes, this chapter can be skipped.

2.1 Markov Chains and Balance Equations

We provide a brief introduction to Markov chains in this section. For a more detailed discussion, refer to Ross (1997), Chapter 4. Consider a stochastic process $\{X_n, n = 0, 1, 2, \ldots\}$ that has finite or countable values. If $X_n = i$, then the process is said to be in state i at time n. Let Ω be the state space of the process. Suppose that whenever the process is in state i, there is a fixed probability, r_{ij}, that it will next be in state j (in the continuous-time version, r_{ij} is the rate of transition from state i to state j), i.e.,

$$P\{X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \ldots, X_1 = i_1, X_0 = i_0\} = r_{ij}$$

for all states $i_0, i_1, \dots, i_{n-1}, i, j$ and all $n \geq 0$. Such a stochastic process is known as a Markov chain. It is well-known that the limiting probability values that the process is in state i, given by π_i, for all states i can be obtained for an irreducible ergodic Markov chain (Ross, 1997) as follows:

π_i is the unique non-negative solution of

$$\pi_i \sum_{j \in \Omega} r_{ij} = \sum_{j \in \Omega} \pi_j \; r_{ji}, \;\; i \in \Omega \text{ and} \sum_{i \in \Omega} \pi_i = 1.$$

This system of equations is commonly called the balance equations. Next, we develop a proof to show that these balance equations can be written as inequalities for any irreducible ergodic Markov chain without changing the unique solution for the limiting probability values. This theorem is later used in the development of an algorithm (BIP) to obtain the exact non-randomized optimal solution for constrained Markov decision processes.

Theorem 2.1.1. *For any ergodic Markov chain with state space Ω, the following systems are equivalent (where Ω_1 is any subset of Ω):*

(I) $$\left\{\pi : \quad \pi_i \times \textstyle\sum_{j \in \Omega}(r_{ij}) \;=\; \sum_{j \in \Omega}\left[\pi_j \times (r_{ji})\right] \quad \forall\, i \;\in\; \Omega.\right\}$$

(II) $$\left\{\pi : \begin{array}{lcll} \pi_i \times \sum_{j \in \Omega}(r_{ij}) & \leq & \sum_{j \in \Omega}\left[\pi_j \times (r_{ji})\right] & \forall\, i \;\in\; \Omega_1 \subseteq \Omega; \\ \pi_i \times \sum_{j \in \Omega}(r_{ij}) & = & \sum_{j \in \Omega}\left[\pi_j \times (r_{ji})\right] & \forall\, i \;\in\; \Omega \backslash \Omega_1. \end{array}\right\}$$

(III) $$\left\{\pi : \begin{array}{lcll} \pi_i \times \sum_{j \in \Omega}(r_{ij}) & \geq & \sum_{j \in \Omega}\left[\pi_j \times (r_{ji})\right] & \forall\, i \;\in\; \Omega_1 \subseteq \Omega; \\ \pi_i \times \sum_{j \in \Omega}(r_{ij}) & = & \sum_{j \in \Omega}\left[\pi_j \times (r_{ji})\right] & \forall\, i \;\in\; \Omega \backslash \Omega_1. \end{array}\right\}$$

In other words, the system of balance equations (I) can be represented by a system of Balance Inequalities (II) or (III): any subset of the equalities in (I) can be replaced by inequalities in the same sense.

Proof. We prove that the systems (II) and (I) are equivalent. The fact that any feasible solution to system (I) is also feasible for system (II) is obvious. The proof of the reverse implication is shown below. Any feasible solution of system (II), π^*, satisfies:

$$\pi_i^* \times \sum_{j \in \Omega}(r_{ij}) \;\leq\; \sum_{j \in \Omega}\left[\pi_j^* \times (r_{ji})\right] \quad \forall\, i \;\in\; \Omega_1 \subseteq \Omega; \tag{2.1}$$

$$\pi_i^* \times \sum_{j \in \Omega}(r_{ij}) \;=\; \sum_{j \in \Omega}\left[\pi_j^* \times (r_{ji})\right] \quad \forall\, i \;\in\; \Omega \backslash \Omega_1. \tag{2.2}$$

Summing the above constraints over all $i \in \Omega$:

$$\sum_{i\in\Omega} \pi_i^* \left[\sum_{j\in\Omega}(r_{ij})\right] \leq \sum_{i\in\Omega}\sum_{j\in\Omega}\left[\pi_j^* \times (r_{ji})\right] = \sum_{j\in\Omega}\sum_{i\in\Omega}\left[\pi_j^* \times (r_{ji})\right] = \sum_{j\in\Omega} \pi_j^* \left[\sum_{i\in\Omega}(r_{ji})\right]$$

$$= \sum_{i\in\Omega} \pi_i^* \left[\sum_{j\in\Omega}(r_{ij})\right]. \tag{2.3}$$

Thus the LHS and the RHS of (2.3), obtained by summing the linear constraints in (2.1) and (2.2), are equal. But since all inequalities in (2.1) are in the same sense, this implies that each inequality has to hold at equality. Therefore any feasible solution of system (II) is also feasible for system (I). Hence, system (II) is identical to system (I). The proof for systems (III) and (I) is identical. □

Notice that (I) is just the system of standard balance equations, whereas (II) and (III) are similar to (I), except that some or all of the balance equations are replaced by less-than-or-equal-to or greater-than-or-equal-to inequalities, respectively. These systems are valid for *any* ergodic Markov chain with finite or infinite state space, in continuous- or discrete-time.

Before we introduce our formulations, we need one additional upper-bounding result:

Corollary 2.1.1. *With respect to system (II), for all states i such that $\sum_{j\in\Omega}(r_{ij}) > 0$,*

$$\pi_i - \frac{\sum_{j\in\Omega}[\pi_j \times (r_{ji})]}{\sum_{j\in\Omega}(r_{ij})} \leq 1$$

is redundant for all $i \in \Omega$.

Proof. The Balance Inequalities in (II) may be rewritten as:

$$\pi_i - \frac{\sum_{j\in\Omega}[\pi_j \times (r_{ji})]}{\sum_{j\in\Omega}(r_{ij})} \leq 0 \tag{2.4}$$

Now the result is immediate. □

An analogous result to Corollary 2.1.1 can be proved for system (III).

2.2 Markov Decision Processes

We provide a brief description of Markov decision processes (MDPs) (Porteus, 2002; Puterman, 1994). We consider a decision maker who is faced with the problem of making decisions that influence the behavior of a probabilistic system as it evolves through time. The goal is to select a sequence of actions that causes the system to perform optimally with respect to some predetermined metric. Since the system we model is ongoing, the state of the system in the future depends on the decisions made in the present. Consequently, the decisions typically must not be made myopically (disregarding the future), but must anticipate the opportunities and costs associated with future system states. We now introduce finite-horizon Markov decision processes. Eventually, we will be interested in infinite-horizon MDPs. These follow from taking limits of the finite-horizon values, typically resulting in stationary behavior (probabilistically independent of time). For more details on this process, refer to Puterman (1994).

Consider a N period MDP. At the beginning of each period, a process is observed to be in some state s, which is a member of the state space S. At each such point in time, an action a must be selected from the set of admissible actions A_s. After the action is chosen, the system evolves probabilistically, in a Markovian manner, to another state, by the beginning of the next period. This means that the next state depends only on the current state and action and not on the history of how the system evolved to the current state. For a discrete-time MDP, the probability of making a transition to state j from state s if action a is chosen is denoted by $\gamma_{j|s,a}$. For a continuous-time MDP, $\gamma_{j|s,a}$ gives the rate at which the system moves to state j from state s if action a is chosen. Define $\Gamma_{s,a} := \sum_{j\in S} \gamma_{j|s,a}$, i.e., it gives the total rate of leaving state s when action a is chosen in that state for a continuous-time MDP, or is equal to one for a discrete-time MDP. The immediate cost resulting from choosing action a in state s is $c_{s,a}$. A decision rule δ is a function defined on the state space that specifies an action for each state. It encapsulates the decision to be made in a given period. A policy $\Pi = (\delta_1, \delta_2, \ldots, \delta_N)$ gives a decision rule for each period. A *non-randomized* policy does not allow any probabilistic mixing of decision rules, while a *randomized* policy does. The objective is to minimize the time-average cost over an infinite-horizon (as $N \to \infty$), i.e., a stationary problem in which the costs, transition probabilities or rates, and state and action spaces do not change over time. Define $\pi_{s,a}$ as the limiting probability value that the system is in state s and action a is chosen.

There exist three algorithms in the literature to solve MDPs: (i) policy iteration, (ii) value iteration and (iii) linear programming. Refer to Puterman (1994) for an excellent comparison of these algorithms. Each of these algorithms gives the optimal non-randomized policy for a given metric to be optimized. We present the general Linear Programming (LP) formulation below. The objective function gives the time-average cost given the selection of actions in each state. The next three systems of equations give the limiting probability values corresponding to a policy (refer the discussion on Markov chains and balance equations in section 2.1). We call the system of equations (2.5) the modified balance equations, since they are a modified version of the balance equations presented earlier.

$$\text{(LP)} \quad \text{MIN} \quad \textstyle\sum_{s\in S} \sum_{a\in A_s} c_{s,a}\ \pi_{s,a}$$

Subject to

$$\sum_{a\in A_j} \Gamma_{j,a}\ \pi_{j,a} - \sum_{s\in S} \sum_{a\in A_s} \gamma_{j|s,a}\ \pi_{s,a} = 0 \quad \text{for all } j \in S \tag{2.5}$$

$$\textstyle\sum_{s\in S} \sum_{a\in A_s} \pi_{s,a} = 1$$

$$\pi_{s,a} \geq 0 \quad \text{for all } s \in S,\ a \in A_s$$

Note that randomized policies are feasible for (LP), but each of the corner points of the (LP) polyhedron correspond to a non-randomized policy. Since the optimal solution obtained by solving linear programs occurs at one of the corner points, it always corresponds to a non-randomized policy.

So far, we have not used any probabilistic constraints, i.e., constraints on some function (we will limit ourselves to linear functions in this dissertation) of the limiting probability values. If probabilistic constraints are included, we call such problems constrained Markov decision processes (CMDPs); we are not aware of any straight-forward implementation of policy iteration or value iteration algorithms to solve CMDPs. The LP algorithm can model such constraints but, typically, results in an optimal randomized policy (Puterman, 1994). The applications that we study in the first part of this dissertation can be modeled as CMDPs, and our focus is to obtain the optimal non-randomized policy for these problems. The only current solution method for this goal is enumeration. We develop algorithms in the following sections of this chapter and conduct computational experiments to determine their efficiency.

2.3 MBEDC Algorithm

Consider any constrained Markov decision process (CMDP) (Puterman, 1994) with state space S and action space A_s in state s. If action $a \in A_s$ is chosen in state s, the cost generated is $c_{s,a}$, and the system transitions to state j with probability/rate $\gamma_{j|s,a}$. Denote by $\Gamma_{s,a}$ the total rate/probability out of state s if action a is chosen, and define $\pi_{s,a}$ as the limiting probability that the system is in state s and action a is selected. The objective is to obtain the optimal non-randomized policy that minimizes the total cost (or maximizes total revenue). A non-randomized policy does not allow any probabilistic mixing of action choices in any state, for example, it does not allow to choose action a_1 with probability p and action a_2 with probability $1-p$ in any state s. Exactly one action must be selected for each state and the selected action must be used every time the system is in that state. Finally, suppose that the solution must satisfy probabilistic constraints of the form, $B\,\pi \leq T$, where B is the left hand side coefficient matrix, T is the right hand side target vector, and $\pi = [\pi_{\mathbf{s,a}}]$ is the vector of variables for the limiting probability values of being in any particular state and choosing action a.

Above, we presented notation for a constrained MDP. We now present additional notation for the MBEDC algorithm formulation. Let z_a^s be a binary variable that takes the value one if action a is chosen in state s, and zero otherwise. Define the set, $Q_k^j := \{(s,a) : s \in S, a \in A_s, \sum_{s,a} \pi_{s,a} = 0 \Leftrightarrow \hat{a}(j) = k\}$, where $\hat{a}(j)$ is the non-randomized action selected in state j. Intuitively, the elements of set Q_k^j are state-action pairs (s,a) such that action a could not have been selected in state s if action j has been selected in state j. Using this notation, the general MBEDC formulation for obtaining the optimal non-randomized policy for a CMDP is:

$$\text{MIN} \quad \textstyle\sum_{s\in S} \sum_{a\in A_s} c_{s,a}\ \pi_{s,a}$$

$$\text{Subject to}$$

$$\textstyle\sum_{a\in A_j} \Gamma_{j,a}\ \pi_{j,a} - \sum_{s\in S} \sum_{a\in A_s} \gamma_{j|s,a}\ \pi_{s,a} = 0 \quad \text{for all } j\ \in\ S$$

$$\textstyle\sum_{s\in S} \sum_{a\in A_s} \pi_{s,a} = 1$$

$$\pi_{s,a} \geq 0 \quad \text{for all } s\ \in\ S,\ a\ \in\ A_s$$

$$B\pi \leq T$$

$$\textstyle\sum_{a\in A_s} z_a^s = 1\ \forall\ s \in S,\ z_a^s \in \{0,1\}\ \forall\ s \in S, \forall\ a \in A_s$$

$$\textstyle\sum_{(s,a)\in Q_k^j} \pi_{s,a} \leq 1 - z_k^j\ \forall\ j \in S,\ \forall\ k \in A_j$$

The objective function gives the time-average cost incurred. The next three systems of equations yield the limiting probability values by solving a system of modified balance equations. The next constraint is the probabilistic constraint we want to model. The second but last system of constraints (binary variables corresponding to a specific state sum upto one for each state) ensures that exactly one action is chosen in each state. The last system of constraints link the binary variables (z_a^s) to the corresponding limiting probability variables (elements of Q_a^s). For example, if z_j^k is one, then all elements of the set Q_j^k are forced to take the value zero, otherwise the constraint is redundant.

We defer further discussion of the MBEDC algorithm and the proof that it yields the optimal non-randomized policy to chapter 3. The computational results for the MBEDC algorithm for the CDOS problem, ADT problem and the multi-item spare parts problems are presented in Sections 3.4.4, 4.1, and 4.2, respectively. The MBEDC algorithm is the first exact and efficient algorithm to obtain the optimal non-randomized policy for any constrained Markov decision process as shown in chapter 3. Also, its relative performance to enumeration is observed to improve as the instance size increases.

2.4 A Different View of Constrained Markov Decision Process

In section 2.2, we refer to the system of equations (2.5) as modified-balance-equations. These reflect a condensed view of solving an MDP, i.e., the limiting probability variables can obtained by solving a system of modified balance equations. A different view of solving a Markov decision process is as follows: If values of all decision parameters are fixed, the resulting system is governed by a Markov chain. Thus, finding the optimal non-randomized policy in a constrained Markov decision process corresponds to finding the optimal combination of the parameter values and hence, the optimal Markov chain whose resulting limiting probability values are feasible for the constraint under question.

Note that decision parameters could be different from the actions to be chosen in each state. A decision parameter may fix the actions in more than one state. For example, if we consider a simple single-threshold based policy for a problem in which a replace action or a do-not-replace action must be chosen at each state. Setting the single threshold parameter to 5, for example, fixes the action in states 1 to 4 as do-not-replace and the action in states 5 onwards as replace. *This selection of the optimal Markov chain from a number of Markov chains can be viewed as solving a disjunctive system of balance equations constraints.* This is what we attempt next.

Consider any constrained Markov decision process (CMDP) with m possible parameter combinations. Assume that the aggregate number of states over all possible parameter choices is v[1], and the v-vector of time-average proportion of time spent in these states is π. Let c be the cost vector associated with these states. Any specific parameter combination l results in a different Markov chain, and hence different system of balance inequalities. We will utilize system (II) throughout; use of system (III) is analogous. Our discussions in the rest of this

[1]We aggregate as some states may not be accessible under all parameter settings.

section continue to hold even if the balance inequalities are replaced by balance equations.

The system of Balance Inequalities given by (2.4) for parameter choice l are denoted by:

$$A^l \; \pi \leq \widehat{0}.$$

In the above $A^l \in R^{v\times v} \; \forall \, l$; $\pi \in R^v$; $\widehat{0} \in R^v$. In addition, following from Corollary 2.1,

$$A^l \; \pi \leq \widehat{1}$$

is a system of redundant constraints ($\widehat{1} \in R^v$). As it is necessary that the time-average proportions of time of any Markov chain sum up to one, we add the constraint $b^l \; \pi = 1$, where $b^l \in R^{1\times v}$ is a 0-1 coefficient vector such that:

$$b^l_j = \begin{cases} 1 & : \text{ if state } j \in \text{the Markov chain induced by parameter setting } l, \\ 0 & : \text{ otherwise.} \end{cases}$$

Note that $b^l \; \pi = 1$ can be equivalently written as: $b^l \; \pi \geq 1$; $\pi \; e = 1$, where e is defined as an e-vector of ones. Thus, the time average proportions of time for the Markov chain induced by the parameter choice l can be obtained by solving:

$$A^l \; \pi \leq \widehat{0}, \quad b^l \; \pi \geq 1, \quad \pi \; e = 1, \quad \pi \geq 0. \tag{2.6}$$

Using the above notation and remembering that the solution must satisfy probabilistic constraints of the form, $B' \; \pi \leq T$ (B' should be obtained from B since we are aggregating states, by introducing zeros in the elements of the matrix B' that correspond to states not originally represented in B), the CMDP problem can be represented as follows:

$$\text{(Q)} \qquad \textbf{Minimize} \quad c\,\pi$$

$$\text{Subject To}$$

$$\left\{ \begin{pmatrix} A^1 \\ -b^1 \end{pmatrix} \pi \leq \begin{pmatrix} \widehat{0} \\ -1 \end{pmatrix} \right\} \vee \left\{ \begin{pmatrix} A^2 \\ -b^2 \end{pmatrix} \pi \leq \begin{pmatrix} \widehat{0} \\ -1 \end{pmatrix} \right\} \vee \ldots \vee \left\{ \begin{pmatrix} A^m \\ -b^m \end{pmatrix} \pi \leq \begin{pmatrix} \widehat{0} \\ -1 \end{pmatrix} \right\}; \tag{2.7}$$

$$B' \; \pi \leq T;$$

$$\pi \; e = 1;$$

$$\pi \geq 0.$$

Thus, the system of equations (2.7) implies that exactly one of the m possible Markov chains corresponding to the m possible parameter combinations should be selected. The limiting probability values of the selected Markov chain must satisfy the probabilistic constraint and must minimize the time-average cost compared to all feasible Markov chains.

2.5 Other Formulations

Balance Equations Disjunctive Programming (BEDP) Formulation.

Using Disjunctive Programming (Balas, 1998), we can obtain an LP formulation for (Q) with balance inequalities being replaced by balance equations. The resulting formulation is shown in (F1) with the first four systems of constraints resulting from the application of the disjunctive programming technique. However, because of the probabilistic constraint, $B\pi \leq T$, ξ_i^1 can be fractional and hence this provides the optimal randomized policy. We add binary constraints as shown in the last but second system of constraints in (F1) to ensure that randomized policies are infeasible. We call this the BEDP formulation; it results in the optimal non-randomized policy.

$$\text{(F1)} \quad \textbf{Minimize} \quad \textstyle\sum_{h=1}^{m} c\, \xi^h$$

$$\text{Subject To}$$

$$A^1\, \xi^1 = \widehat{0}, \quad A^2\, \xi^2 = \widehat{0}, \quad \ldots\ , \quad A^m\, \xi^m = \widehat{0};$$

$$-b^1\, \xi^1 = -\xi_0^1, \quad -b^2\, \xi^2 = -\xi_0^2, \quad \ldots\ , \quad -b^m\, \xi^m = -\xi_0^m;$$

$$B\, \xi^1 \leq \xi_0^1 T, \quad B\, \xi^2 \leq \xi_0^2 T, \quad \ldots\ , \quad B\, \xi^m \leq \xi_0^m T;$$

$$\xi^1\, e = \xi_0^1, \quad \xi^2\, e = \xi_0^2, \quad \ldots\ , \quad \xi^m\, e = \xi_0^m,$$

$$\textstyle\sum_{i=1}^{m} \xi_0^i = 1, \; \xi_0^i \in \{0,1\} \; \forall i,$$

$$(\xi^i, \xi_0^i) \geq 0 \qquad i = 1, \ldots, m.$$

If $\xi_0^j = 1$, then the system of balance equations that corresponds to Markov chain j is activated and the limiting probability values of that Markov chain are obtained. However, if $\xi_0^j = 0$ then all limiting probability values corresponding to Markov chain j are forced to take the value zero. Thus, the optimal non-randomized policy is obtained, since exactly one of all the ξ_0^i can be one and the rest must be zero.

Balance Inequalities Disjunctive Programming (BIDP) Formulation.

This formulation is similar to the previous one, except we use Balance Inequalities instead of Balance Equations. It also results in the optimal non-randomized solution. The explanation is same as in the previous case.

$$\text{(F2)} \quad \textbf{Minimize} \quad \textstyle\sum_{h=1}^{m} c\ \xi^h$$

$$\text{Subject To}$$

$$A^1\ \xi^1 \le \widehat{0}, \quad A^2\ \xi^2 \le \widehat{0}, \quad \ldots\ , \quad A^m\ \xi^m \le \widehat{0};$$

$$-b^1\ \xi^1 \le -\xi_0^1, \quad -b^2\ \xi^2 \le -\xi_0^2, \quad \ldots\ , \quad -b^m\ \xi^m \le -\xi_0^m;$$

$$B\ \xi^1 \le \xi_0^1 T, \quad B\ \xi^2 \le \xi_0^2 T, \quad \ldots\ , \quad B\ \xi^m \le \xi_0^m T;$$

$$\xi^1\ e = \xi_0^1, \quad \xi^2\ e = \xi_0^2, \quad \ldots\ , \quad \xi^m\ e = \xi_0^m,$$

$$\textstyle\sum_{i=1}^{m} \xi_0^i = 1, \ \ \xi_0^i \in \{0,1\}\ \forall i,$$

$$(\xi^i, \xi_0^i) \ge 0 \qquad i = 1, \ldots, m.$$

Balance Inequalities Integer Programming (BIP) Formulation.

Next, we show how to obtain an MIP formulation for disjunctions of Balance Inequality systems as in (Q) using Integer Programming (Nemhauser and Wolsey, 1988). For every Markov chain l (or more precisely, every Markov chain corresponding to a particular choice of parameter settings l), we introduce a binary variable y^l such that:

$$y^l = \begin{cases} 1 & : \ \text{if Markov chain } l \text{ is selected,} \\ 0 & : \ \text{otherwise.} \end{cases}$$

Below, is the BIP formulation which also yields the optimal non-randomized policy for any CMDP. Note that we need the balance inequalities result for this formulation, otherwise the first system of constraints in (F3) will be infeasible if balance equations are used. The explanation of formulation (F3) is as follows: The objective gives the time-average cost incurred. The second type of constraint in (F3) requires all the binary variables to sum to one, results in the selection of one binary variable with value one, and the others are forced to be zero. If $y^k = 1$, then the balance equations of Markov chain k are activated (refer the first row of constraints in (F3)), and the balance equations corresponding to all other Markov chains $i \neq k$ become redundant (refer the first row of constraints in (F3) and Corollary 2.1.1). Thus, exactly one Markov chain is selected and hence the BIP algorithm gives the optimal non-randomized policy.

$$\text{(F3)}\quad \textbf{Minimize} \quad c\,\pi$$

Subject To

$$\left\{\begin{pmatrix} A^1 \\ -b^1 \end{pmatrix} \pi \leq \begin{pmatrix} \widehat{1} - ey^1 \\ -y^1 \end{pmatrix}\right\}, \left\{\begin{pmatrix} A^2 \\ -b^2 \end{pmatrix} \pi \leq \begin{pmatrix} \widehat{1} - ey^2 \\ -y^2 \end{pmatrix}\right\}, \ldots \left\{\begin{pmatrix} A^m \\ -b^m \end{pmatrix} \pi \leq \begin{pmatrix} \widehat{1} - ey^m \\ -y^m \end{pmatrix}\right\};$$

$$\textstyle\sum_{l=1}^{m} y^l = 1;$$

$$B\pi \leq T, \quad \pi\, e = 1, \quad \pi \geq 0, \quad y \in \{0,1\}^m.$$

Results of computation time of the BEDP and BIP formulations are shown in Tables 2.1 and 2.2 for two MDP problems: the dynamic service capacity problem, which is closely related to the CDOS problem (chapter 3) without any service constraints, and for the classical (s, S) inventory problem. We observed that the performance of the BIDP algorithm is dominated by the performance of the BEDP algorithm (the BIDP algorithm is a weaker version of the BEDP algorithm) and hence we only discuss the BEDP and the BIP algorithms. Again, we defer the detailed discussion of the MBEDC algorithm to the next chapter. We also observe that neither of these three formulations are efficient for these two problems, i.e., they are slower than enumeration. (We do not discuss these test problems in detail as they are not problems studied in the dissertation.)

We use CPLEX 6.6 Mixed Integer Program (MIP) solver on a Sun Ultra 60 Model 2360 machine with a 360MHz UltraSPARC-II processor to conduct our computational experiments. The BIP algorithm visits each and every node of the branch and bound tree and thus enumerates over all Markov chains, but it also solves additional LP subproblems generated by the branch and bound algorithm. Thus, it is not efficient compared to pure enumeration. The BEDP algorithm is not efficient because of the block structure of its constraints. Each block corresponds to a particular Markov chain and since there are no constraints that link the different blocks, it is typically faster to solve the individual blocks, i.e., enumerate. However, the spare parts application that we study in chapter 4 has a constraint that links the different blocks, and it may be interesting to test the performance of the BEDP algorithm for this problem. The close-to-optimal performance of the greedy heuristic for this problem, and the non-degeneracy (we discuss degeneracy in chapter 4) of the BEDP formulation suggest that the BEDP algorithm may yield better than expected results for computation time.

Algorithm	Number of parameter (n) choices						
	5	10	20	40	60	80	101
BEDP (avg)	2.07	3.17	3.20	8.99	21.67	32.62	36.71
BIP (avg)	3.00	3.95	7.23	18.36	49.79	120.65	177.97
Enumeration (avg)	0.05	0.1	0.2	0.4	0.6	0.8	1.01
BEDP (max)	3.47	4.17	4.90	23.97	33.17	70.24	67.01
BIP (max)	4.34	5.68	21.83	50.39	89.61	249.59	278.09

Table 2.1: Computation time results (in seconds) for DSC problem.

Algorithm	Number of parameter choices (s, S)					
	(3,3)	(5,5)	(10,10)	(15,15)	(20,20)	(25,25)
BEDP (avg)	0.10	0.34	2.03	5.36	9.52	22.95
BIP (avg)	0.29	3.64	26.71	111.37	264.32	782.65
Enumeration (avg)	0.09	0.25	1	2.25	4	6.25
BEDP (max)	0.11	0.37	2.77	7.02	12.13	28.48
BIP (max)	0.39	4.46	27.55	128.2	299.87	831.31

Table 2.2: Computation time results (in seconds) for (s, S) problem.

2.6 Short Summary

Until now the only exact solution method to obtain the optimal non-randomized policy for a constrained Markov decision process has been enumeration. We present four different formulations based on Integer Programming and Disjunctive Programming to find an exact and efficient algorithm for this problem. While three of these formulations were not efficient in our experiments, we show in the next chapter that the MBEDC algorithm is indeed efficient. We also show that the balance equations of a Markov chain that yield the limiting probability values can also be written as balance inequalities without changing the unique limiting probability values of the Markov chain.

In the next chapter, we look at the constrained dynamic operator staffing problem, which we define as the problem of determining the optimal number of permanent and temporary operators in a call center, and the threshold values of when to use the temporary operators, so as to minimize the time-average hiring and opportunity costs subject to certain service level constraints. This problem is of interest because call center managers are facing increasing pressure to reduce costs while at the same time ensuring an acceptable level of service quality. We model this problem as a constrained Markov decision process and solve it using the MBEDC algorithm developed in this chapter.

Chapter 3

The Constrained Dynamic Operator Staffing Problem

For many organizations in the service industry, call center quality is a critical component of customer loyalty and hence revenue generation. Call centers offer a product or service through telephone lines (or the Internet) to customers, who in turn have expectations regarding the quality of the service they will receive. These expectations can be classified into two categories: time spent waiting for an operator to provide service, and the human-interaction with the operator. To improve the human-interaction element of service, call centers provide training to their staff or operators. To reduce customer waiting time, call centers can hire more (or better) operators. But, while the need to meet service level goals is critical, call centers also face increasing pressure to reduce costs. Thus, call center managers are concerned with increasing the *efficiency* of call centers (Gans et al., 2003) - improving service quality while controlling costs. One way to achieve staffing efficiency is to correctly schedule operators: Over-staffing leads to unnecessary costs, while under-staffing will result in dissatisfied customers and possibly a loss of business/revenue (Brigandi et al., 1994).

Focus. We develop a mathematical model to determine an optimal non-randomized operator staffing schedule; a schedule, not involving probabilistic mixing, that minimizes total cost for a call center with service level goals. These goals can be naturally modeled as constraints that any feasible operator staffing level must satisfy, which can be incorporated into a mathematical program. Despite this fact, this work is the first to use such a formulation to find an optimal non-randomized schedule. Typical examples of call center service level constraints are: (i) The probability of no delay is above P, (ii) the average waiting time is less than T time units, (iii) the average number of customers in queue is less than Q, etc. Opportunity costs associated with these service level goals can also be incorporated into the objective function.

Complicating the operator staffing problem is the fact that the rate of customer arrival to the call center may be time-varying; it may be risky to develop a call center model that does not take this into account (Gans et al., 2003). Specifically, if the number of operators (determined using the forecast of the average arrival rate) is fixed for each period, the following two scenarios may occur: (i) The arrival rate may be lower than expected, resulting in operators being idle and hence, unnecessary costs, or (ii) the arrival rate may be higher than expected, resulting in long waiting times and hence, inability to meet the call center service level goals. To counter these two scenarios, and in fact even under stationary arrival rates, call centers may benefit from *flexible* staffing - the ability to dynamically adjust staffing levels with traffic. Such flexibility may be attained by utilizing temporary operators as a complement to the permanent operators who are always available to provide service. The temporary operators may be either supervisors/managers or other operators who are on call (Whitt 1999, Jongbloed and Koole 2001); these temporary operators provide service at the call center manager's discretion. Note that another way to cope with time varying arrival rate is to alter the staffing levels, without using temporary operators, by breaking the day into different blocks or periods. This must be proactive, not reactive. In addition, even with such fixed schedules temporary operators could still be utilized for each block.

Since our operator staffing problem includes service level constraints and dynamic arrival and service rates, we call this problem the Constrained Dynamic Operator Staffing (CDOS) problem. The CDOS problem involves determining the number of permanent operators to hire, the number of temporary operators to hire, and the number of temporary operators to use at every state of the call center queuing model, in order to minimize the time-average hiring and opportunity cost subject to the service level constraints. For simplicity, we restrict our attention to threshold policies for utilizing the temporary operators. (These may, in general, be sub-optimal.) Threshold policies specify two critical values: (i) If the number of jobs in system reaches the higher critical value, the call center manager asks the temporary operators to provide service. (ii) When the number of jobs in system falls to the lower threshold value and all temporary operators are idle, the temporary operators stop providing service. This restriction to threshold policies is similar to, and shares common motivation with, the restriction to base-stock policies in complex inventory environments. Examples are models for spare parts networks (Wong et al., 2006), perishable items (Deniz et al., 2005, and the references therein), and supply chains featuring multiple supply modes (Veeraraghavan and Scheller-Wolf, 2006).

Under suitable assumptions on the arrival process and service time distributions, the CDOS problem can be modeled as a Markov Decision Process (MDP) with probabilistic constraints (Puterman, 1994). For MDPs *without* constraints, there exist several efficient solution algorithms (Puterman, 1994; Bertsekas, 1995; Porteus, 2002), namely policy iteration, value iteration and linear programming. However, if there are probabilistic constraints, we are not aware of any straightforward implementation of policy iteration or value iteration. The existing linear programming method can model such constraints, but typically results in an optimal *randomized* policy (Puterman, 1994). A randomized threshold policy for the CDOS problem corresponds to using a chance mechanism between two or more threshold values, for example 40% of the time the manager should ask the temporary operators to provide service if the number of jobs in system reaches ten, and 60% of the time he should wait until the number of jobs in system reaches fifteen. As mentioned above, we focus on finding the optimal *non-randomized* threshold policy for the CDOS problem; non-randomized policies are in general more prevalent and easier to implement than randomized policies. The only exact solution method in the literature for obtaining the optimal non-randomized policy for constrained MDPs is enumeration, which is computationally prohibitive for problems of any significant size.

Research questions and contributions. We answer the following research questions for the CDOS problem in this chapter.

- How can one model the CDOS problem?

Section 3.1 describes the Constrained Dynamic Operator Staffing (CDOS) problem. It explains the hiring and opportunity costs used in the model. We discuss the threshold policies for the CDOS problem and show how to model the CDOS problem as a constrained Markov decision process.

- Is there an exact and efficient algorithm to obtain the optimal non-randomized threshold policy values for the CDOS problem?

Section 3.2 presents the existing Linear Program (LP) method that results in an optimal randomized threshold policy solution for the CDOS problem. We develop the Modified Balance Equations Disjunctive Constraints (MBEDC) algorithm to obtain an optimal non-randomized solution for the CDOS problem in Section 3.3. This yields a Mixed Integer Program (MIP) formulation (Nemhauser and Wolsey, 1988) which can be solved quickly by a commercial solver. We bridge, for the first time, the two areas of integer programming (Nemhauser and Wolsey, 1988) and constrained Markov decision process (Puterman, 1994; Altman, 1999) by developing an exact and efficient algorithm to obtain optimal non-randomized policies. Computational results featuring computational speed are discussed in Section 3.4. We observe that the computation times of the MBEDC algorithm are lower than enumeration by up to a factor of 200 and by a factor of 10 on average, and that the MBEDC algorithm's relative performance improves with increasing instance size.

- How much are the economic savings from hiring temporary operators? How does the maximum number of temporary operators available effect the total cost? What is the economic cost of satisfying service level constraints?

Computational results highlighting economic insights for managers are discussed in Section 3.4 for the service level constraints corresponding to the probability that an arriving customer does not have to wait for service, and limits on the average number in queue. We observe when a corresponding opportunity cost associated with the constraint is included in the model, the actual service level constraint is redundant up to a certain target level. Nevertheless, our experiments show that the average decrease in cost from employing temporary operators is 20%-40% for the former service level constraints and 30%-45% for the latter service level constraint. We also show that either with or without temporary operators, the economic cost of satisfying service level constraints can be considerable. Finally, we show that staffing flexibility not only reduces costs but also eliminates staffing inefficiencies in the presence of service level constraints, i.e., with only permanent operators the service level constraint may be over-satisfied, while with temporary and permanent operators, the service level constraint is satisfied exactly.

- Can this algorithm be extended to obtain the optimal non-randomized policy for any constrained MDP?

The MBEDC algorithm can be applied to any constrained MDP; this is discussed in detail in Section 3.3. We discuss extensions of the CDOS model in Section 3.5, provide two additional specific applications in chapter 4, and present conclusions and directions for future work in Section 7.2.

Related Literature. In this paper we consider a call center operator staffing problem with service level constraints. In addition, we account for time-varying customer arrival rates and allow the number of operators to be increased dynamically in response to system congestion and the customer arrival rate. The inclusion of these three factors simultaneously in an operator staffing problem is novel. Such an objective and similar constraints are proposed in Jongbloed and Koole (2001) also, but they consider the problem of providing stochastic guarantees for constraint satisfaction by considering percentile levels of an arrival-rate random variable. Another difference is that they call upon the temporary operators when the realized *non-varying* arrival-rate in a period is higher than expected, while in our model the arrival rate, and hence the optimal number of operators, may change stochastically *within* a scheduling period.

There is a vast literature dealing with the problem of long-term operator staffing to meet time-varying demand. In general refer for example to Hall (1991), Jennings et al. (1996), Gans et al. (2003), and the references therein. However, these papers do not consider service level constraints.

3.1 CDOS Problem Description

We model a call center with both permanent and temporary operators (supervisors or stand-by operators). We assume the permanent operators are always available to provide service, but the call center manager decides when to use the temporary operators. We define the system state as the number of jobs in system, and initially we assume that the maximum number of jobs in the system, N, is finite resulting in a finite state space. (We address models with infinite state space in Section 3.5.) In each state, the optimal number of temporary operators to be used may be different; however, such policies are more complicated to implement and analyze than simple threshold policies. Therefore we restrict our focus to threshold policies for temporary operators.

In a threshold policy, the call center manager asks all of the temporary operators to provide service when the number of jobs in system increases to a threshold value. The temporary operators continue to provide service until the number of jobs in system reaches a second (typically lower) threshold value, and all temporary operators are idle. When this happens, we assume the temporary operators stop providing service en masse and return to stand-by mode. We also assume that if a temporary operator is providing service to a customer, the customer will not be transferred to a permanent operator if one becomes idle. We make this assumption for practical considerations, as a customer may be dissatisfied if the operator is changed during service. Finally, we assume that when a customer arrives and both a permanent operator and a temporary operator are idle, the customer is assigned to the permanent operator. We can model and solve the cases when *any* of the above assumptions are relaxed.

We seek to minimize the time-average hiring and opportunity cost subject to service level constraints. The following subsections provide details of the cost structure and of the service level constraints that we consider.

3.1.1 Costs and Decision Parameters

Let the number of permanent operators hired be x and the number of temporary operators hired be y. The time-average cost of hiring each permanent (temporary) operator is c_1 (c_2), with $c_2 < c_1$; it is cheaper to hire temporary operators on stand-by than to have full-time permanent operators. (The $c_2 \geq c_1$ case results in $y = 0$.) Initially, only the permanent operators provide service and the call center behaves as a x-server central queue system. Let the two threshold values for changing the number of operators be denoted by n and m ($n > m$). Thus when the number of jobs in system reaches n, all y temporary operators are asked to provide service and the call center behaves as a $(x + y)$-server central queue system (permanent and temporary operators may have different service rates as explained in Section 3.1.2). Once all y temporary operators are idle and the number of jobs in system is at or below m, the temporary operators return to stand-by mode. In this paper we assume m to be fixed at x for simplicity, but in general m can also be a decision variable. In addition to the fixed cost of hiring one temporary operator, c_2, there may be a variable cost, c_3, per serving temporary operator incurred only for the proportion of time that temporary operator is providing service. In the case of a supervisor acting as a temporary operator, c_3 can be considered as a penalty cost since the supervisor delays her original work in order to provide service. Note that once the temporary operators start providing service, the number of jobs in system can be less than x but one or more temporary operators may still be providing service, as calls are not transferred during service. Thus, we need to keep track of the number of temporary operators actually providing service, and not just available, at each state of the system because of the marginal cost c_3.

In addition to the operator hiring costs, there may also be opportunity costs related to the service level goals. We define d as the one-time cost incurred if a customer experiences positive delay, and w as the opportunity cost per unit time incurred for each customer waiting in queue. We include these costs in our model as traditionally service level goals in call center optimization problems have been modeled as opportunity costs rather than service level constraints (example: Andrews and Parsons, 1993). Modeling service level goals as constraints, while more accurate (d and w are usually estimates), makes the problem more difficult to solve. We discuss solution issues in Section 3.2.

3.1.2 Arrival Process and Service Time Models

We use a two-state Markov Modulated Poisson Process (MMPP) to represent the time-varying customer arrival process. For details on 2-state MMPP arrival process, refer to Nain and Núñez-

Queija (2001). When the state of the system is i, the customer arrival process transitions between a Poisson process with "low" arrival rate λ_1 and a Poisson process with "high" arrival rate λ_2 ($\lambda_1 < \lambda_2$) with exponential transition rates $\alpha(i)$ and $\beta(i)$ respectively, as shown in Figure 3.1. Thus, the transition rates between the two arrival rates can be state-dependent.

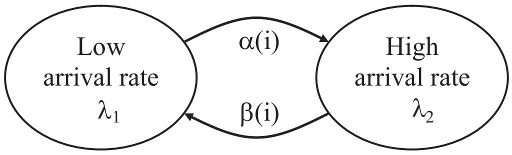

Figure 3.1: 2-State MMPP arrival process.

This model implies that the threshold value n (Section 3.1.1) is actually a vector of two components, i.e., there are two threshold values n_1 and n_2 corresponding to the low arrival-rate process and the high arrival-rate process, respectively. Thus, if the system is operating under the low (high) arrival-rate process, the call center manager will ask the temporary operators to provide service when the number of jobs in system reaches the threshold value n_1 (n_2). This raises an important question - how will a call center manager determine if the system is in the low/high arrival-rate process? One approach is to look at the customer arrival pattern over a time interval of certain length to determine the current arrival rate. There are other approaches as well, but we do not elaborate on these in this paper, and simply assume that the call center manager knows the instantaneous rate. We assume customer service times are exponentially distributed with rate μ_1 when a permanent operator is providing service, and rate μ_2 when a temporary operator is providing service; these rates may or may not be equal. We discuss extensions of these models to k-state MMPP arrival processes as well as more general BMAP arrival processes (Lucantoni, 1993), and/or phase-type service distributions (Osogami and Harchol-Balter, 2003) in Section 3.5.

Since we seek to minimize time-average cost we need a necessary condition for system stability in the general infinite state case, in order for the time-average cost to be finite. The instantaneous load of a system at time t, $\rho(t)$, is defined as the ratio of the customer arrival rate at time t, $\lambda(t)$, to the service rate at time t, $\mu(t)$. Within our dynamic setting, $\lambda(t)$ and $\mu(t)$ are random variables. We assume that it is possible to choose adequate numbers of permanent and temporary operators ($\hat{x}$ and $\hat{y}$, respectively), and hence adequate service capacity ($\hat{\mu} = \hat{x}\ \mu_1 + \hat{y}\ \mu_2$), to ensure that the system is stable over an infinite horizon, i.e., $\hat{\mu} > \lambda_2$. We do allow for transient overload; $\rho(t) > 1$ is permitted for some but not all t.

3.1.3 CDOS Problem as a Markov Decision Process with Constraints

An important result of our model assumptions is that once we fix (i) the number of permanent operators, x, (ii) number of temporary operators y, and (iii) the threshold values n_1 and n_2, the call center system can be represented by a single Markov chain. Therefore, for fixed values of x and y the CDOS problem without any service level constraints (only minimizing the time-average hiring and opportunity cost) can be represented as a Markov Decision Process (MDP) under an average reward criterion (Puterman, 1994) with n_1 and n_2 as the decision parameters. We discuss the action choices for such an MDP in this section, and defer the discussion on rewards (or costs in the case of the CDOS problem) and transition rates to Section 3.2. But first, we provide an example in Figure 3.2.

This example corresponds to a call center with $x = 2$, $y = 2$, $n_1 = 4$, $n_2 = 4$, $\mu_1 = \mu_2 = \mu$, $\alpha(i) = \alpha$, and $\beta(i) = \beta$ for all i. The state ua (ub) implies that the number of jobs in system is u, the arrival process is Poisson with rate λ_1 (λ_2), and only the permanent operators are providing service. The state uc (ud) implies that the number of jobs in system is $u(\geq x + y)$, the arrival process is Poisson with rate λ_1 (λ_2), and all x permanent operators and y temporary operators are providing service. Finally, state u_1, u_2a (u_1, u_2b) implies that the number of jobs in system is $u_1 + u_2(< x + y)$, the arrival process is Poisson with rate λ_1 (λ_2), and $u_1(\leq x)$ permanent operators and $u_2(\leq y)$ temporary operators are providing service.

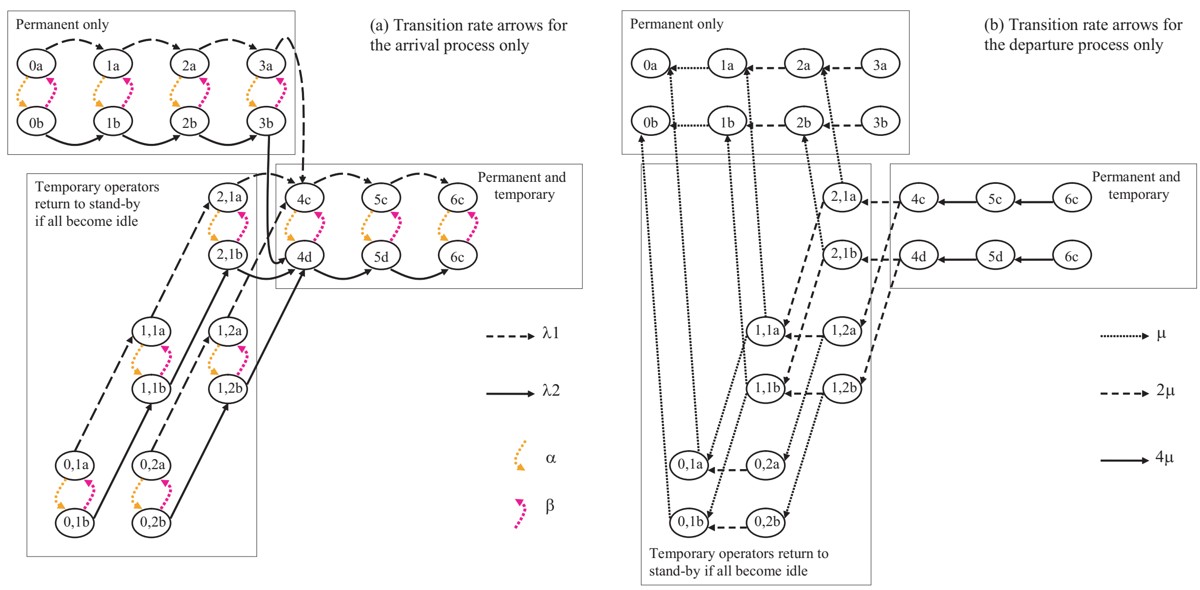

Figure 3.2: CDOS problem - Markov chain representation when $x = 2, y = 2, n_1 = n_2 = 4, \mu_1 = \mu_2 = \mu, \alpha(i) = \alpha, \beta(i) = \beta, N = 6$.

The situation where $n_1 > n_2$ (example: 5 and 4, respectively) is more delicate. Refer to Figure 3.3 for a Markov chain with the same parameters as Figure 3.2, except $n_1 = 5$ and n_2

= 4. In this case, when $n_1 > n_2$, we call the states where the arrival process is "low" and the number of jobs in system is greater than or equal to n_2 but less than n_1 as "between-off" states. If the system is in a "between-off" state (example: 4a in Figure 3.3), temporary operators are not called. We model them remaining on stand-by even if the arrival process transitions from the low arrival-rate process to the high arrival-rate process, at which time the number of jobs is at or above n_2. We call the resulting states as "between-on" states (example: 4b in Figure 3.3). Thus, we assume for simplicity that in the "between-on" states, the manager will call in the temporary operators only after there is an arrival. (We can model and solve the case where if the system enters a "between-on" state, the manager immediately calls in the temporary operators. This results in a MDP that is significantly more complicated.) We model the situation where $n_2 > n_1$ similarly.

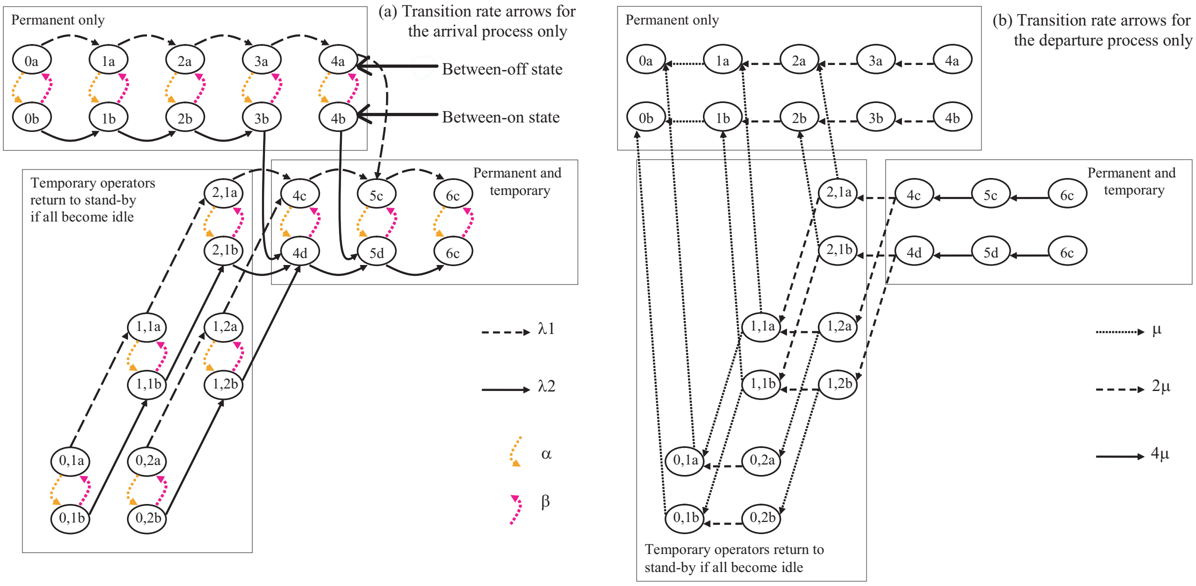

Figure 3.3: CDOS problem - Markov chain representation when $x = 2, y = 2, n_1 = 5, n_2 = 4, \mu_1 = \mu_2 = \mu, \alpha(i) = \alpha, \beta(i) = \beta, N = 6$.

The action state space for the MDP can be best explained by referring to Figure 3.2. At each state ua (ub), there are two action choices: (i) Ask the temporary operators to help if there is an arrival (a_1), or (ii) continue without their help (a_2). There is only one pre-determined action choice (all operators are available) for each of the remaining states (uc, ud, u_1, u_2a, and u_1, u_2b). If we select action a_1 in state $\hat{u}$a ($\hat{v}$b) and there is an arrival ($\hat{u} = \hat{v} = 3$ in Figure 3.2), the Markov chain transitions to state $[\hat{u}$+1]c ($[\hat{v} + 1]d$). Alternatively, if action a_2 is selected in state $\hat{u}$a ($\hat{v}$b) and there is an arrival ($\hat{u} = \hat{v} = \{0, 1, 2\}$ in Figure 3.2), the Markov chain transitions to state $[\hat{u}$+1]a ($[\hat{v} + 1]b$).

For fixed values of x and y, the MDP can be solved to obtain optimal threshold values of n_1

and n_2 that minimize time-average cost. For the case without service level constraints, there exist three efficient solution techniques in the literature, and Puterman (1994) is an excellent reference for their description and comparison: (i) Policy iteration, (ii) value iteration, and (iii) linear programming. Note that we need to fix the values of x and y as we do not allow there to be different values for these parameters at different states of the system, which may happen if we allow these to be action choices also. Solving the resulting MDPs for all choices of x and y, and then selecting values of x, y, n_1 and n_2 that minimize the time-average cost will yield the optimal threshold policy.

However, the CDOS problem has service level constraints which complicate things significantly; for example: (i) The probability of no delay is above P, (ii) the average waiting time is less than T time units, (iii) the average number of customers in queue is less than Q, etc. Constraints (ii) and (iii) are related by Little's law and hence we only consider constraints (i) and (iii). The CDOS problem is thus an MDP problem with probabilistic constraints for fixed values of x and y. There is no straightforward implementation of policy iteration or value iteration for such problems (Puterman, 1994), but linear programming may be applied to problems of this form. We discuss the linear programming algorithm in the next section.

3.2 Linear Programming Method and Randomized Policies

Let $S^{x,y}$ be the state space of the MDP for the CDOS problem for fixed x and y, and $A_s^{x,y}$ be the set of action choices in state $s \in S^{x,y}$. Also, we define $S_1^{x,y}$ ($S_2^{x,y}$) $\subset S^{x,y}$ as follows: States corresponding to the low (high) arrival-rate process such that all operators in that state are busy constitute the set $S_1^{x,y}$ ($S_2^{x,y}$). Let $L^{x,y}$ and $H^{x,y}$ be the set of states in which there is at least one idle operator (temporary operators do not count unless they are available for service) corresponding to the low arrival rate process and the high arrival rate process, respectively. We define $\pi_{s,k}$ to be the limiting probability that the MDP is in state s and the call center manager chooses action k; $c_{s,k}$ captures any costs associated with choosing action k in state s. Also, we define $n_q(s)$ as the number of customers in queue in state s, which is equal to the difference between the number of jobs in system and the number of operators that are busy in state s, and $y(s)$ is defined as the number of temporary operators providing service (actually busy and not just available) in state s. Finally, $\Gamma_{s,k}$ is the total transition rate out of state s if action k is selected, and $\gamma_{j|s,k}$ is the rate of transition to state j if action k is selected in state s. The general Linear Program (LP) formulation for the CDOS problem without constraints is shown below. For a discussion of this LP, refer to Puterman (1994), page 391. The solution of this LP yields the optimal action choices in each state that minimizes the time-average cost.

$$\text{(LP)} \qquad \text{MIN} \sum_{s \in S^{x,y}} \sum_{k \in A_s^{x,y}} c_{s,k}\ \pi_{s,k}$$

Subject to

$$\sum_{k \in A_j^{x,y}} \Gamma_{j,k}\ \pi_{j,k} - \sum_{s \in S^{x,y}} \sum_{k \in A_s^{x,y}} \gamma_{j|s,k}\ \pi_{s,k} = 0 \quad \text{for all } j\ \in\ S^{x,y}$$

$$\sum_{s \in S^{x,y}} \sum_{k \in A_s^{x,y}} \pi_{s,k} = 1$$

$$\pi_{s,k} \geq 0 \quad \text{for all } (s,k) \text{ such that } s\ \in\ S^{x,y},\ k\ \in\ A_s^{x,y}$$

Below in ($\mathcal{LP}$), we provide the LP formulation for the CDOS problem with the delay constraint, the constraint for average number of customers in queue, and the time-average cost function explicitly written as a function of the cost structure defined in Section 3.1.1.

First, we discuss the objective function. The first term corresponds to the time-average opportunity cost of delay that is incurred only when a customer arrives and finds all operators busy. The second term is the time-average opportunity cost incurred for having customers wait in queue. The third (fourth) term is the cost of hiring permanent (temporary) operators and finally, the fifth term is the marginal time-average cost incurred when the temporary operators are providing service. We call the system of constraints (3.1) Modified-Balance-Equations, which in conjunction with (3.2) and (3.5) yield the limiting probability values. Constraints (3.3) and (3.4) correspond to the service level constraints for delay and average number of customers in queue, respectively. Note that constraint (3.3) is an ensemble average, rather than a time average. It is derived by conditioning on the arrival process as follows:

$$\text{P[arrival sees no delay]} = \text{P[arrival is in low] P[arrival sees no delay|low]}$$

$$+ \text{P[arrival is in high] P[arrival sees no delay|high]}$$

$$= \frac{\frac{\lambda_1}{\alpha}}{\frac{\lambda_1}{\alpha}+\frac{\lambda_2}{\beta}} \left(\frac{\sum_{s\in L^{x,y}} \sum_{k\in A_s^{x,y}} \pi_{s,k}}{\frac{\frac{1}{\alpha}}{\frac{1}{\alpha}+\frac{1}{\beta}}} \right) + \frac{\frac{\lambda_2}{\beta}}{\frac{\lambda_1}{\alpha}+\frac{\lambda_2}{\beta}} \left(\frac{\sum_{s\in H^{x,y}} \sum_{k\in A_s^{x,y}} \pi_{s,k}}{\frac{\frac{1}{\beta}}{\frac{1}{\alpha}+\frac{1}{\beta}}} \right)$$

$$= \frac{(\alpha+\beta)\lambda_1}{\beta\lambda_1+\alpha\lambda_2} \sum_{s\in L^{x,y}} \sum_{k\in A_s^{x,y}} \pi_{s,k} + \frac{(\alpha+\beta)\lambda_2}{\beta\lambda_1+\alpha\lambda_2} \sum_{s\in H^{x,y}} \sum_{k\in A_s^{x,y}} \pi_{s,k}.$$

Constraints (3.5) ensure that the limiting probability variables cannot have negative values.

$$(\mathcal{LP}) \quad \text{MIN} \;\; d \left(\lambda_1 \sum_{s\in S_1^{x,y}} \sum_{k\in A_s^{x,y}} \pi_{s,k} + \lambda_2 \sum_{s\in S_2^{x,y}} \sum_{k\in A_s^{x,y}} \pi_{s,k} \right) + w \sum_{s\in S^{x,y}} \sum_{k\in A_s^{x,y}} n_q(s)\, \pi_{s,k} +$$

$$c_1\, x + c_2\, y + c_3 \sum_{s\in S^{x,y}} \sum_{k\in A_s^{x,y}} y(s)\, \pi_{s,k}$$

Subject to

$$\sum_{k\in A_j^{x,y}} \Gamma_{j,k}\, \pi_{j,k} - \sum_{s\in S^{x,y}} \sum_{k\in A_s^{x,y}} \gamma_{j|s,k}\, \pi_{s,k} = 0 \quad \text{for all } j \in S^{x,y} \tag{3.1}$$

$$\sum_{s\in S^{x,y}} \sum_{k\in A_s^{x,y}} \pi_{s,k} = 1 \tag{3.2}$$

$$\frac{(\alpha+\beta)\lambda_1}{\beta\lambda_1+\alpha\lambda_2} \sum_{s\in L^{x,y}} \sum_{k\in A_s^{x,y}} \pi_{s,k} + \frac{(\alpha+\beta)\lambda_2}{\beta\lambda_1+\alpha\lambda_2} \sum_{s\in H^{x,y}} \sum_{k\in A_s^{x,y}} \pi_{s,k} \geq P \tag{3.3}$$

$$\sum_{s\in S^{x,y}} \sum_{k\in A_s^{x,y}} n_q(s)\, \pi_{s,k} \leq Q \tag{3.4}$$

$$\pi_{s,k} \geq 0 \quad \text{for all } (s,k) \text{ such that } s \in S^{x,y},\ k \in A_s^{x,y} \tag{3.5}$$

If there are no service goals, i.e., constraints (3.3) and (3.4) are not included in the model, $(\mathcal{LP})$ becomes a form of (LP), which, even though the linear program does not enforce non-randomized policies, yields the optimal non-randomized stationary distribution and cost. This is because without (3.3) and (3.4) the vertices of the resulting LP polyhedron have a one-to-one correspondence with non-randomized policies. Since one optimal solution to an LP always corresponds to one of the vertices of the polyhedron, an optimal non-randomized policy can be found. Moreover, the optimal values of n_1 and n_2 can then be inferred from the limiting probability values (refer to Puterman, 1994, page 393).

It is important to note that one of the drawbacks of this linear programming formulation (LP or $\mathcal{LP}$) is that the solution does not directly provide the optimal action in each state. These have to be deduced from the limiting probability variables that are strictly positive; the limiting probability variables have a one-to-one correspondence with state-action pairs, and in the optimal non-randomized solution only one of the limiting probability variables corresponding to a particular state will be strictly positive. One can then solve the resulting MDPs for each combination of x and y values to obtain the globally optimal solution.

If service goals need to be considered, i.e., constraints (3.3) and/or (3.4) *are* included (such constraints are referred to as probabilistic constraints in the literature), ($\mathcal{LP}$) typically yields a *randomized* optimal policy (refer Puterman, 1994, page 406). This means that in some states it may be optimal to use a chance mechanism to determine the course of action. An example of a randomized policy in the CDOS problem is: If the call center system is in the low arrival-rate process, 40% of the time the manager should ask the temporary operators to provide service if the number of jobs in system reaches ten, and 60% of the time he should wait until the number of jobs in system reaches fifteen. This randomization occurs because some of the vertices of the polyhedron defined by ($\mathcal{LP}$) correspond to randomized policies, in particular those vertices where at least one of the constraints (3.3) or (3.4) is binding. It is important to note that in this case deducing the action selection in each state is more difficult, as more than one limiting probability variable corresponding to a state will be positive: Additional steps are required to obtain the actual randomization probabilities/parameter(s) (solving a system of equations), and it is known that, in general, it is *very difficult* to find the optimal randomization parameter(s). For example, in Shwartz and Makowski (1990) the optimal randomization parameter is obtained as a unique root of an explicit function in (0, 1), and in Nain and Ross (1986) it is obtained by using an online algorithm. For more results on constrained Markov decision processes see Altman (1999). Also, while such randomized policies may be practical in certain environments, they are, in general, harder to implement in practice than non-randomized policies.

Currently, the only known method for obtaining an exact non-randomized solution for MDPs with constraints is enumeration, which is often computationally prohibitive. For example, in a CDOS problem with 20 choices each for x, y, n_1 and n_2, enumeration will need to solve (obtain limiting probability values and the objective value) 20^4 (=160000) Markov chains. In contrast to this, we provide a novel, exact and efficient approach to solve problem instances in the next section.

3.3 Modified Balance Equations Disjunctive Constraints Algorithm

We propose the Modified Balance Equations Disjunctive Constraints (MBEDC) algorithm to obtain the optimal *non-randomized* policy for the CDOS problem. First, fix x and y to obtain an MDP with constraints with decision parameters n_1 and n_2. Let I_1 be the set of choices for n_1 and I_2 be the set of choices for n_2. Define $z_i^{n_1}$ and $z_l^{n_2}$ as binary variables as follows: (Note that the superscripts n_1 and n_2, respectively, indicate the two decision parameters and are not variable indices of these binary variables. The subscripts i and l, respectively, are variable indices.)

$$z_i^{n_1} = \begin{cases} 1 & : \quad n_1 = i \\ 0 & : \quad n_1 \neq i \end{cases} \text{ for all } i \in I_1, \qquad z_l^{n_2} = \begin{cases} 1 & : \quad n_2 = l \\ 0 & : \quad n_2 \neq l \end{cases} \text{ for all } l \in I_2.$$

Also, define $Q_i^{n_1} = \{(s,k) : s \in S^{x,y},\ k \in A_s^{x,y},\quad \sum_{(s,k)} \pi_{s,k} = 0 \Leftrightarrow n_1 = i\}$ for all $i \in I_1$. Similarly, $Q_l^{n_2} = \{(s,k) : s \in S^{x,y},\ k \in A_s^{x,y},\quad \sum_{(s,k)} \pi_{s,k} = 0 \Leftrightarrow n_2 = l\}$ for all $l \in I_2$. (Note that the superscripts n_1 and n_2, respectively, again indicate the two decision parameters and are not variable indices of these sets. The subscripts i and l, respectively, are variable indices.) Intuitively, $Q_i^{n_1}$ ($Q_l^{n_2}$) is the set of state-action pairs such that if any element of $Q_i^{n_1}$ ($Q_l^{n_2}$) has strictly positive limiting probability mass then it implies that n_1 (n_2) is not equal to i (l) and vice versa. For example, recalling from Section 3.1.3 that a_1 is the action that the manager will ask all the temporary operators to start providing service if there is an arrival and a_2 is the action that she will not, $Q_{10}^{n_2} = \{(0b, a_1),\ (1b, a_1),\ (2b, a_1),\ \ldots,\ (8b, a_1),\ (9b, a_2),\ (10b, a_2),\ \ldots,\ ([n_2^{max} - 1]b, a_2)\}$, where n_2^{max} is the maximum allowable value of n_2. State-action pairs $(0b, a_1)$, $(1b, a_1)$, $(2b, a_1)$, $\ldots$, $(8b, a_1)$ are in $Q_{10}^{n_2}$, because if action a_1 is selected in any of these states then it contradicts n_2=10. Also, n_2=10 implies selection of action a_1 in state $9b$ and hence, $(9b, a_2)$ is in $Q_{10}^{n_2}$. Finally, $(10b, a_2)$, $\ldots$, $([n_2^{max} - 1]b, a_2)$ are in $Q_{10}^{n_2}$, because of our assumption that temporary operators will be asked to start service in these "between-on" states if there is an arrival (refer Section 3.1.3). Note that every $Q_l^{n_2}$, $l \in I_2$, contains either (jb, a_1) or (jb, a_2) for all $j \in I_2$. Similarly, every $Q_i^{n_1}$, $i \in I_1$, contains either (ja, a_1) or (ja, a_2) for all $j \in I_1$.

We now present the Mixed Integer Programming formulation ($\mathcal{IP}$) that gives the optimal *non-randomized* policy.

$$(\mathcal{IP}) \quad \text{MIN} \;\; d \left(\lambda_1 \sum_{s \in S_1^{x,y}} \sum_{k \in A_s^{x,y}} \pi_{s,k} + \lambda_2 \sum_{s \in S_2^{x,y}} \sum_{k \in A_s^{x,y}} \pi_{s,k} \right) \; + \; w \sum_{s \in S^{x,y}} \sum_{k \in A_s^{x,y}} n_q(s) \; \pi_{s,k} \; +$$

$$c_1 \; x \; + \; c_2 \; y \; + \; c_3 \sum_{s \in S^{x,y}} \sum_{k \in A_s^{x,y}} y(s) \; \pi_{s,k}$$

Subject to

$$\sum_{k \in A_j^{x,y}} \Gamma_{j,k} \; \pi_{j,k} - \sum_{s \in S^{x,y}} \sum_{k \in A_s^{x,y}} \gamma_{j|s,k} \; \pi_{s,k} = 0 \quad \text{for all } j \; \in \; S^{x,y} \tag{3.6}$$

$$\sum_{s \in S^{x,y}} \sum_{k \in A_s^{x,y}} \pi_{s,k} = 1 \tag{3.7}$$

$$\frac{(\alpha+\beta)\lambda_1}{\beta\lambda_1 + \alpha\lambda_2} \sum_{s \in L^{x,y}} \sum_{k \in A_s^{x,y}} \pi_{s,k} + \frac{(\alpha+\beta)\lambda_2}{\beta\lambda_1 + \alpha\lambda_2} \sum_{s \in H^{x,y}} \sum_{k \in A_s^{x,y}} \pi_{s,k} \geq P \tag{3.8}$$

$$\sum_{s \in S^{x,y}} \sum_{k \in A_s^{x,y}} n_q(s) \; \pi_{s,k} \leq Q \tag{3.9}$$

$$\pi_{s,k} \geq 0 \quad \text{for all } (s,k) \text{ such that } s \; \in \; S^{x,y}, \; k \; \in \; A_s^{x,y} \tag{3.10}$$

$$\sum_{i \in I_1} z_i^{n_1} = 1, \qquad z_i^{n_1} \in \{0,1\} \quad \text{for all } i \; \in \; I_1 \tag{3.11}$$

$$\sum_{l \in I_2} z_l^{n_2} = 1, \qquad z_l^{n_2} \in \{0,1\} \quad \text{for all } l \; \in \; I_2 \tag{3.12}$$

$$\sum_{(s,k) \in Q_i^{n1}} \pi_{s,k} \leq 1 - z_i^{n_1} \text{ for all } i \; \in \; I_1, \quad \sum_{(s,k) \in Q_l^{n2}} \pi_{s,k} \leq 1 - z_l^{n_2} \text{ for all } l \; \in \; I_2 \tag{3.13}$$

The objective function and the constraints (3.6), (3.7), (3.8), (3.9), and (3.10) are the same as in the $(\mathcal{LP})$ formulation in Section 3.2. Constraints (3.11) and (3.12) ensure that the optimal solution will select exactly one value each for n_1 and n_2 and hence, randomized policies are infeasible. Finally, constraints (3.13) link the binary variables to the appropriate continuous limiting probability variables. If $z_i^{n_1}$ ($z_l^{n_2}$) equals one, then all limiting probability variables corresponding to the state-action pairs in set $Q_i^{n_1}$ ($Q_l^{n_2}$) are forced to take the value zero. If

$z_i^{n_1}$ ($z_l^{n_1}$) is zero, the constraint becomes redundant. These linking constraints are also known as big-M constraints in Integer Programming (Nemhauser and Wolsey, 1988); for this case, $M = 1$. Next, we show that MBEDC yields the optimal threshold policy out of all non-randomized threshold policies for the CDOS problem. The proof can be extended to any constrained MDP problem.

Theorem 3.3.1. *The MBEDC algorithm yields the optimal non-randomized threshold policy for the CDOS problem.*

Proof. Any non-randomized threshold (we omit the word threshold from now on) policy for the CDOS problem results in a single choice for n_1 and n_2 each; thus, in each state of the MDP, exactly one action is selected. The resulting system is then governed by the corresponding Markov chain. The limiting probability values of the Markov chain corresponding to any such non-randomized policy that satisfies the service level constraints (3.8)-(3.9) yield a feasible solution for $(\mathcal{IP})$. Thus, any non-randomized solution to the CDOS problem also lies within the solution space of $(\mathcal{IP})$.

We now show that the solution space of $(\mathcal{IP})$ contains only non-randomized policies. It is known (Puterman, 1994) that any feasible solution to (3.6)-(3.10) corresponds to some possibly randomized policy, i.e., one or more actions may be selected with fixed probability in some states of the MDP. We need to show that the limiting probability values obtained from any strictly randomized policy are infeasible for $(\mathcal{IP})$. Assume there exists a state s such that $\pi_{s,a_1} > 0$ and $\pi_{s,a_2} > 0$. Without loss of generality, assume s corresponds to a low arrival rate state. Since every $Q_i^{n_1}$, $i \in I_1$, contains either (ja, a_1) or (ja, a_2) for all $j \in I_1$, $\pi_{s,a_1} > 0$ and $\pi_{s,a_2} > 0$ implies that $n_1 \neq i$ for all $i \in I_1$. Thus, (3.13) forces $z_i^{n_1} = 0$ for all $i \in I_1$, violating (3.11). Thus, any feasible solution to $(\mathcal{IP})$ must correspond to a non-randomized policy. This also implies that the indices of the binary variables that have value one in the optimal solution to $\mathcal{IP}$ yield the optimal threshold values; $z_i^{n_j} = 1 \Leftrightarrow n_j^* = i$. Finally, the objective function of $(\mathcal{IP})$ gives the time-average cost incurred for any feasible non-randomized policy. Thus, for given x and y, $(\mathcal{IP})$ yields the optimal non-randomized policy. □

Below we summarize a few key facts about the MBEDC algorithm:

- System (3.13) is a system of disjunctive constraints, which gives rise to the name Modified Balance Equations Disjunctive Constraints (MBEDC) algorithm.
- The solution of the LP relaxation of $(\mathcal{IP})$ is obviously the optimal randomized policy. Thus, this algorithm can yield both the optimal randomized and optimal non-randomized policies.

- The resulting MIP formulations for each combination of x and y can be solved quickly using any commercial solver and the optimal values of x, y, n_1, and n_2 can thus be obtained.
- The optimal values of n_1 and n_2 can be deduced directly from the binary variables and do not need to be inferred from the limiting probability variables; the indices of the two binary variables (one for each threshold parameter) that have value one in the optimal solution immediately yield the optimal threshold values.
- The MBEDC algorithm can be applied to *any* constrained MDP. Obviously, the choice of binary variables is problem dependent; for a given problem, introduce one binary variable (z_i^k) for each possible choice (indexed by i) of each decision parameter (indexed by k). The binary variables corresponding to each parameter should sum to 1 (as in (3.11) and (3.12)), i.e., $\sum_i \; z_i^k = 1 \quad \forall \; k$. This ensures that each decision parameter takes exactly one value from all possible choices. Finally, sets Q_i^k must be defined and their respective elements should be appropriately chosen. The MIP should include the linking constraints (as in (3.13)), i.e. $\sum_{s \in Q_i^k} \; \pi_s \; \leq \; 1 - z_i^k \quad \forall \; i, \; \forall \; k$.
- In chapter 4, we apply the MBEDC algorithm to two other problems. The first involves obtaining the optimal threshold values for a threshold-based resource allocation policy for a multiserver queuing system. In particular, we analyze the adaptive dual threshold (ADT) policy proposed in Osogami et al. (2005) and compare its performance to the general state dependent optimal policy. The second problem is a multi-item, two location spare parts problem with lateral transshipments, with the objective of finding the optimal non-randomized policy so as to minimize total inventory costs subject to waiting time constraints at each location, see Wong et al. (2006).

We examine the performance of $(\mathcal{IP})$ for the CDOS problem in the next section.

3.4 Computational Results

In this section we provide computational results for (i) the economic analysis of the CDOS problem, which is now possible due to the MBEDC algorithm and (ii) the computational speed of the MBEDC algorithm. We use CPLEX 6.6 Mixed Integer Program (MIP) solver on a Sun Ultra 60 Model 2360 machine with a 360MHz UltraSPARC-II processor to conduct our computational experiments. In particular, we choose the primal method to solve the (Linear Program) sub-problems. All other settings of the MIP solver are selected at their default value.

Refer to Table 3.1 in Section 3.4.4 for details on the different choices of the parameters for the experiments in this section. Throughout we focus on the probability of no delay and number in queue service level constraints, individually. Typically, when combined one is redundant. The target probability that a customer must be served immediately (P) is varied from 0.1 to 0.3 in steps of 0.05. The target expected number of customers in queue (Q) is chosen from the set: $\{1, 2, 4, 8, 16, 32\}$. Similar experiments can be carried out to obtain insights specific to other service level constraints. While we limit ourselves to a maximum of twenty permanent and five temporary operators in these experiments, we can solve for higher values of these parameters.

3.4.1 Economic Savings of Hiring Temporary Operators

We conduct experiments to determine the economic savings of using temporary operators by solving the CDOS problem with and without temporary operators, and comparing the respective optimal objective values. For each ratio of c_1 to (c_2+c_3), the average percentage decrease in costs from using temporary operators for the probability of no delay constraint and the number in queue constraint are shown in Figure 3.4a and Figure 3.4b, respectively.

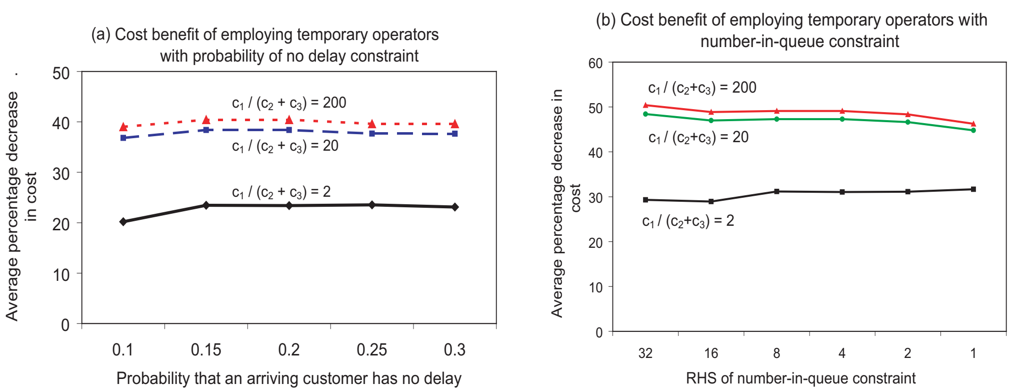

Figure 3.4: Economic savings of employing temporary operators with service level constraints.

First, look at Figure 3.4a for the results for the no delay constraint. Over all instances in our experiments, the decrease in costs from hiring temporary operators ranged from 0% to 80%. Keeping the ratio of c_1 to $(c_2 + c_3)$ fixed, the average percentage decrease in cost from using temporary operators changes by a maximum of 2%-3% as P is varied from 0.1 to 0.3. One explanation for this consistency is that keeping all other parameters fixed as P is increased from 0.1 to 0.3, the number of permanent operators required if no temporary operators are allowed increases by at most one. When temporary operators are allowed, some of the permanent operators are replaced by the temporary operators resulting in lower costs while satisfying the service level constraint at the same time.

Next, look at Figure 3.4b for the results for the number in queue constraint. Over all instances in our experiments, the actual percentage decrease in costs from hiring temporary operators ranged from 0% to 83% for this constraint. The average percentage decrease in cost from using temporary operators for this constraint appears to change by a maximum of only 7-8% when the right hand side of the constraint is varied from 32 to 1. Compared to the probability of no delay constraint, the average percentage decrease in cost from using temporary operators is higher for this constraint. One possible explanation is as follows. For this constraint without temporary operators, as the right hand side increases (the constraints becomes less strict), the optimal number of permanent operators required to satisfy the constraint decreases by one, until the constraint is not binding as a result of the corresponding opportunity cost. We discuss this issue in section 3.4.3 of this chapter. If temporary operators are allowed, (maximum of five) permanent operators are replaced by temporary operators. In addition, adding one (permanent or temporary) operator results in a relatively higher decrease in the expected number in queue compared to the decrease in probability of no delay (for example: expected number in queue may decrease from 8 to 6, a decrease of 2, while the probability of no delay may decrease from 0.3 to 0.25, a decrease of 0.05) because the temporary operator is utilized in the tail of the queue in such cases. Thus, for the same values of d and w, there is a higher decrease in the opportunity cost of number in queue compared to the opportunity cost of no delay. These two factors result in high percentage savings from hiring plus opportunity costs for the number in queue constraint as compared to the probability of no delay constraint.

We also find, as expected, that when the cost of using temporary operators compared to the cost of using permanent operators becomes cheaper (the ratio of c_1 to $c_2 + c_3$ increases) for a fixed value of P or Q, the percentage decrease in costs for both cases increases. But as the operator cost ratio is increased beyond a certain critical level, there is no further increase in the percentage decrease in costs. This is because we limit the maximum number of temporary operators that can be used; once the optimal number of temporary operators to be used reaches this maximum limit the costs no longer decrease. This indicates a need for increasing the maximum number of temporary operators available to the system.

Note that it is now also possible to obtain the expected value of perfect information (EVPI) of the real time arrival rate. While the two optimal threshold values (n_1 and n_2) are obtained from the optimal solution, how can the call center manager determine what the true customer arrival rate is in real time: Is it λ_1 or λ_2? We calculated the percentage increase in cost from forcing the call center manager to use $n_1 = n_2$ compared to the base case. This gives us the

EVPI of the real-time arrival rate. In all our experiments, the maximum EVPI value is less than 0.4%. This is a very surprising result. It says that the call center manager should invest no (or negligible) resources in trying to determine the real-time arrival rate. The main concern should be to determine the correct staffing levels of the permanent and temporary operators and possibly just a single threshold value for the number in queue at which the temporary operators are asked to provide service. This is true because the high arrival rate threshold dominates. If in both low and high arrival rates, the call center manager is forced to use the same threshold n, then typically n will be very close or equal to n_2, and the system will then either not reach the threshold often during the low arrival process, or if it does, it will not spend much time in the states where the temporary operators provide service. Thus, forcing n_1 and n_2 to be equal results in only a small increase in the total cost, as the temporary operators are most important in the high arrival states. We see similar results in a related application that we study in chapter 4, i.e., adding more thresholds to such call center models does not significantly effect the optimal cost.

We now move to studying some specific experimental instances to gain further, more detailed insights.

3.4.2 Effect of Maximum Number of Temporary Operators Available

Given the effect of our limit on the number of temporary operators in Section 3.4.1, we now study the effect of the maximum number of temporary operators available on the reduction in cost for different values of P and Q, respectively.

Consider the probability-of-no-delay constraint. We fix $c_1 = 1$, w=0, Nt=50, $\alpha(i) = \beta(i)$=5, $\mu_1 = \mu_2$=2 in Figure 3.5. For the plots corresponding to (a), (b) and (c) ((d), (e) and (f)), we choose λ_1=6 (18) and λ_2=9 (27). We vary the opportunity cost of delay (d) between 0.01, 1, and 10. (Results for $d = 0$ and $d = 0.1$ are similar to those for $d = 0.01$). When d=0.01, the hiring costs dominate the opportunity costs, and if d=10 then the reverse is true.

First, refer to Figures 3.5a, 3.5b, and 3.5c corresponding to λ_1=6 and λ_2=9. When the hiring costs dominate the opportunity costs (d=0.01), adding up to four temporary operators decreases costs as each temporary operator is able to replace a permanent operator while satisfying the constraint. Adding the fifth temporary operator is useful only when $P = 0.3$, because only up to five permanent operators are required for all lower values of P, and hence the fifth temporary operator cannot replace a permanent operator for these values of P. As

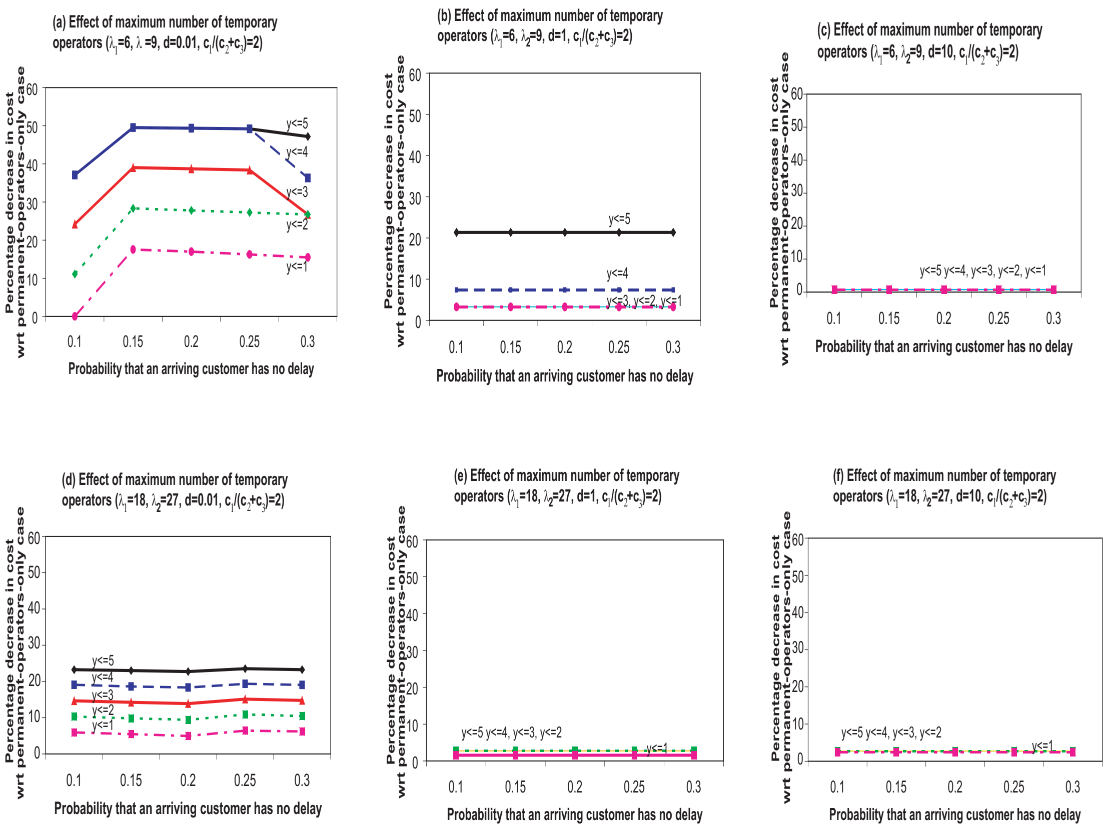

Figure 3.5: Effect of maximum number of temporary operators on cost reduction (c_2=0.1, c_3=0.4).

opportunity cost increases (d=1), more permanent operators are required in general, and the ability of the temporary operators to replace the permanent operators is diminished because of opportunity cost considerations. For this case (d=1), one to three temporary operators are able to replace only one permanent operator, but even using three temporary operators compared to one does not reduce the opportunity cost enough to outweigh their hiring costs. However, if four or five temporary operators are available they replace more permanent operators and hence result in a greater reduction in cost. Finally, as the opportunity cost dominates ($d = 10$), a high number of permanent operators are hired to reduce the opportunity cost and the temporary operators are unable to replace any permanent operator without significantly increasing the opportunity cost. Thus, at most one temporary operator is useful and aids only in reducing the opportunity cost. Hence, as the opportunity cost increases the percentage reduction in costs from using temporary operators decreases but is not negligible.

Similar observations hold for the case when λ_1=18 and λ_2=27 (Figures 3.5d, 3.5e, and 3.5f), except that the percentage reduction in costs from using temporary operators first decreases and then increases as the opportunity cost is increased. (The percentage decrease in cost is

2.46% in Figure 3.5f vs. 1.46% in Figure 3.5e for $y \leq 1$.) The reason is that we have limited the maximum number of permanent operators to twenty and for the busier call center more permanent operators are needed. Since more permanent operators are not available temporary operators are used to reduce the opportunity cost. Thus, temporary operators can help reduce costs if opportunity costs are high and not enough permanent operators are available. From these observations for the "less busy" and "more busy" call centers we see that assessing the magnitude of the opportunity cost is crucial.

Next, consider the number in queue constraint. We fix $c_1 = 1$, d=0, Nt=50, $\alpha(i) = \beta(i)$=5, $\mu_1 = \mu_2$=2 in Figure 3.6. For the plots (a), (b) and (c) ((d), (e) and (f)), we choose λ_1=6 (18) and λ_2=9 (27). We vary the opportunity cost of number in queue (w) between 0.01, 1, and 10. (Results for $w = 0$ and $w = 0.1$ are similar to those for $w = 0.01$). When w=0.01, the hiring costs dominate the opportunity costs, and if w=10 then the reverse is true. Figure 3.6 considers the case when the ratio of c_1 to $c_2 + c_3$ is 2. Each temporary operator provides additional benefit in this case by replacing a permanent operator, except when the opportunity cost w dominates (=10). In that situation it is beneficial to employ up to three temporary operators. The additional cost of adding a fourth temporary operator is not compensated by the corresponding drop in opportunity cost nor can it replace a permanent operator while still satisfying the constraint. We observe that the results for the two constraints are similar. As mentioned before, adding one (permanent or temporary) operator results in a relatively higher decrease in the expected number in queue compared to the decrease in probability of no delay (for example: expected number in queue may decrease from 16 to 14, a decrease of 2, while the probability of no delay may decrease from 0.65 to 0.6, a decrease of 0.05). Thus, for the same values of d and w, there is a higher decrease in the opportunity cost of number in queue compared to the opportunity cost of no delay. This explains the higher, and more sustained, percentage decrease values for the number in queue constraint.

3.4.3 Economic Cost of Satisfying Service Level Constraints

The economic cost of satisfying any service level goal (constraint) can be obtained by comparing the optimal objective values of the CDOS problem with and without that particular constraint, respectively. We study the percentage increase in costs as a result of increasing the right hand side value of our service level constraints from their base value. The percentage increase in cost for the probability of no delay constraint when c_1=1, c_2=0.1, c_3=0.4, Nt=50, $\alpha(i) = \beta(i)$=5, $\mu_1 = \mu_2$=2 are shown in Figure 3.7 and Figure 3.8 for the cases when no temporary operators

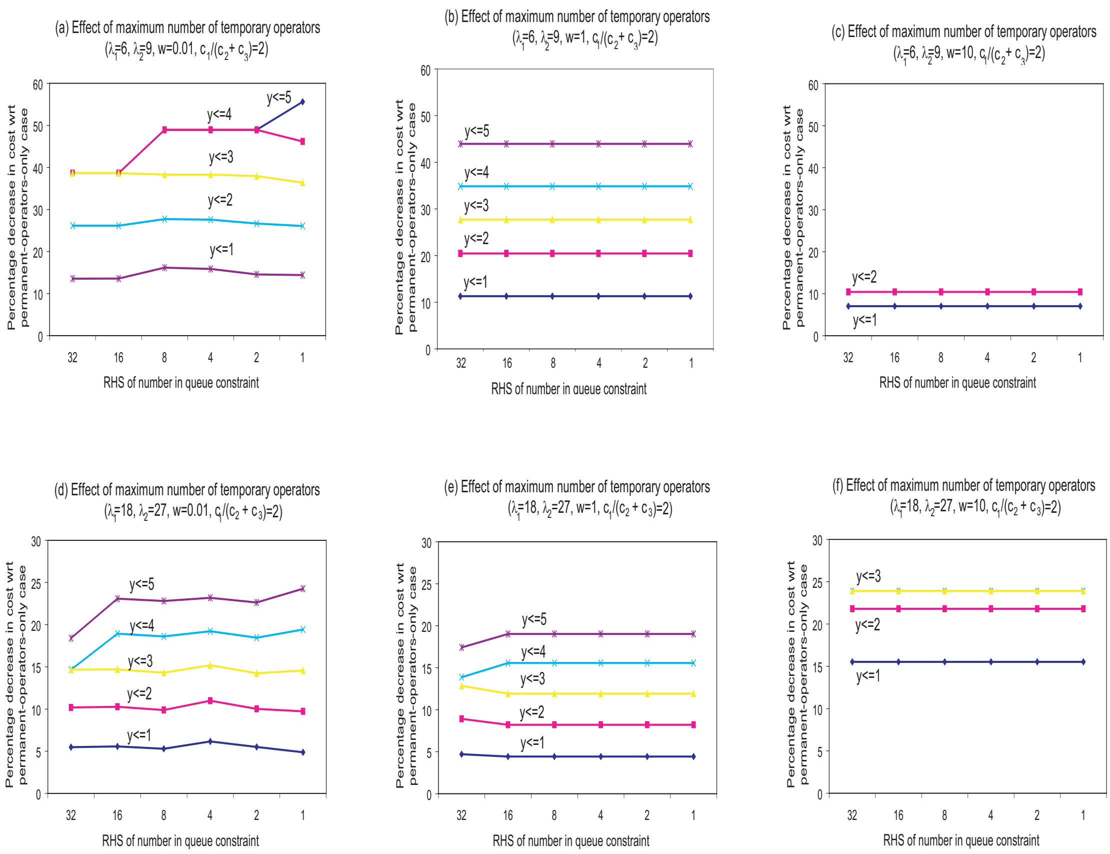

Figure 3.6: Effect of maximum number of temporary operators on cost reduction (p_2=0.1, p_3=0.4)

are allowed and when temporary operators up to five are allowed, respectively, for different values of d (0, 0.01, 0.1, 1, and 10).

Figure 3.7 and Figure 3.8 show that for a given value of d, the delay constraint is redundant up to a certain value of P, $\hat{P}(d)$, which is increasing in d, and hence there is no increase in costs up to this value (for example, in the left plot of Figure 3.7, $\hat{P}(d) = 0.1$ for $d \leq 0.1$ and $\hat{P}(d) \geq 0.3$ for $d \geq 1$). As expected the percentage cost of satisfying the delay constraint is increasing in P, keeping all other parameters fixed. One feature of note from Figure 3.7 are the "plateaus" in the increase in cost. Refer to the points marked A, B and C, which correspond to using four, five and five permanent operators, respectively when d=0. An increase in P from 0.1 to 0.15 caused the optimal solution to increase the number of permanent operators by one. Ideally, if we have five permanent operators, we would prefer to be at point C, where the highest possible service level target for probability of no delay is satisfied with five permanent operators. At point B, keeping five permanent operators on staff over-satisfies the constraint, and wastes money. Now looking at Figure 3.8, we see that the "plateaus" are greatly reduced, and the cost increases

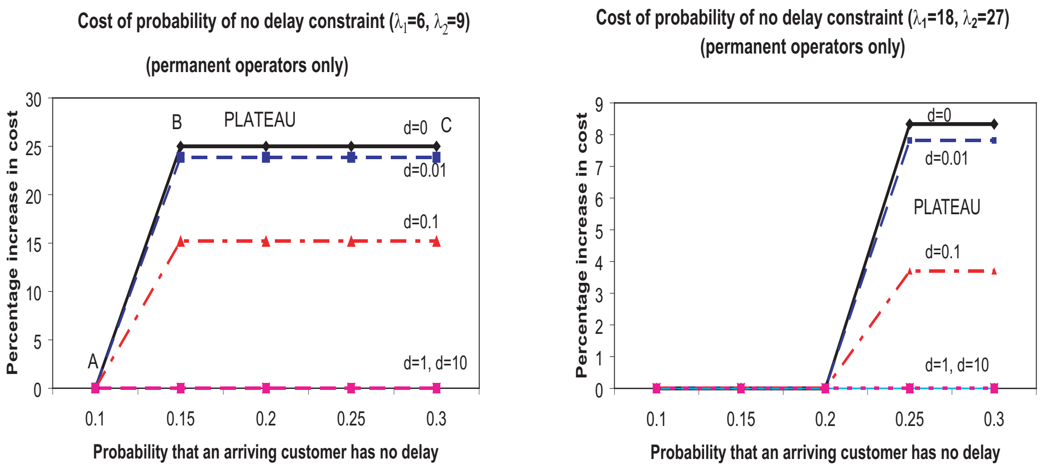

Figure 3.7: Economic cost of satisfying the delay constraint with permanent operators only. $\mu_1 = \mu_2 = 2, \alpha(i) = 0.5, \beta(i) = 0.5, Nt = 50, w = 0, c_1 = 1, c_2 = 0.1, c_3 = 0.4$.

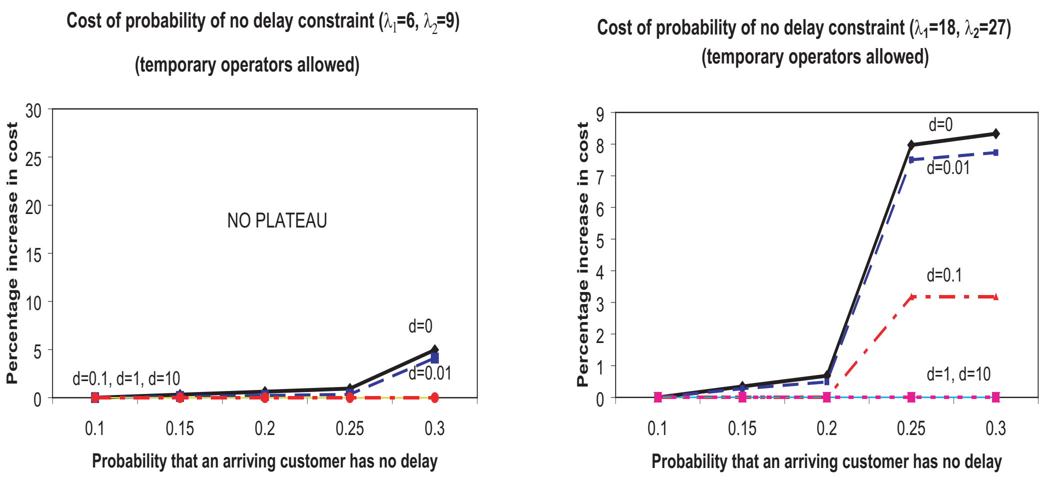

Figure 3.8: Economic cost of satisfying the delay constraint with up to five temporary operators. $\mu_1 = \mu_2 = 2, \alpha(i) = 0.5, \beta(i) = 0.5, Nt = 50, w = 0, c_1 = 1, c_2 = 0.1, c_3 = 0.4$.

more gradually. The same increase in P from 0.1 to 0.15 for d=0 results in one permanent operator and four temporary operators, or a savings of 49.5% compared to the case without temporary operators. In fact, for this case the number of permanent operators increased to five (plus temporary operators) only at $P = 0.7$. This means that staffing inefficiencies at points like B can be avoided by using temporary operators. This highlights a key benefit of flexible staffing in the presence of service level constraints.

Looking at both Figure 3.7 and Figure 3.8, we see the delay constraint may have significant impact on the costs of the system (maximum percentage increase in costs are 25% and 9%, respectively). We obtain similar trends for different settings of the input parameter values. The main takeaways from Figure 3.7 and Figure 3.8 are thus that hiring temporary operators may have two advantages: (i) the obvious advantage is that it may be cheaper (if $c_2 < c_1$) and hence reduces costs while maintaining efficiency, and (ii) more subtly, provides "finer control" to achieve staffing efficiency while satisfying service constraints.

The "plateau" effect is even more pronounced in the number in queue constraint case as shown in Figures 3.9 and 3.10. Similar comments as above for the probability of no delay case hold here as well. Finally, similar results to the case when $c_1/(c_2+c_3)$=2 are obtained for all studies when this ratio is changed to 20 and 200.

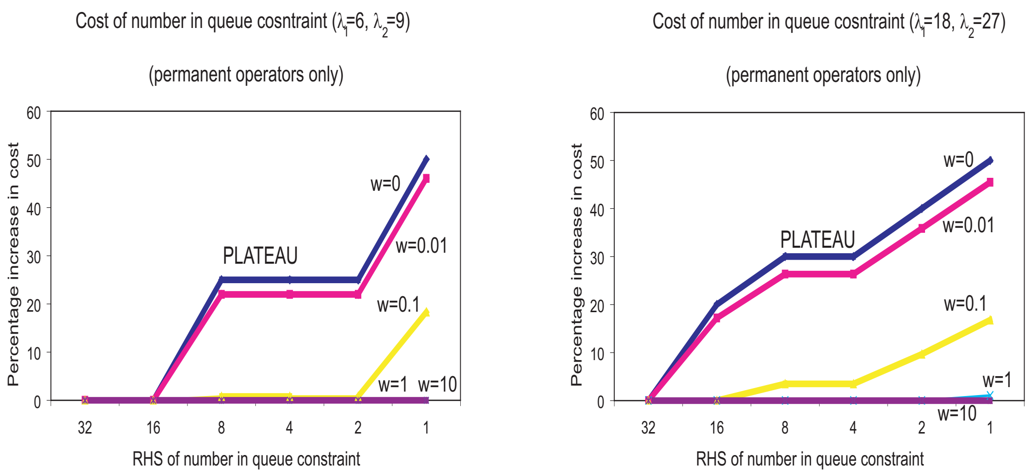

Figure 3.9: Economic cost of satisfying the delay constraint with permanent operators only. $\mu_1 = \mu_2 = 2, \alpha(i) = 0.5, \beta(i) = 0.5, Nt = 50, d = 0, c_1 = 1, c_2 = 0.1, c_3 = 0.4.$

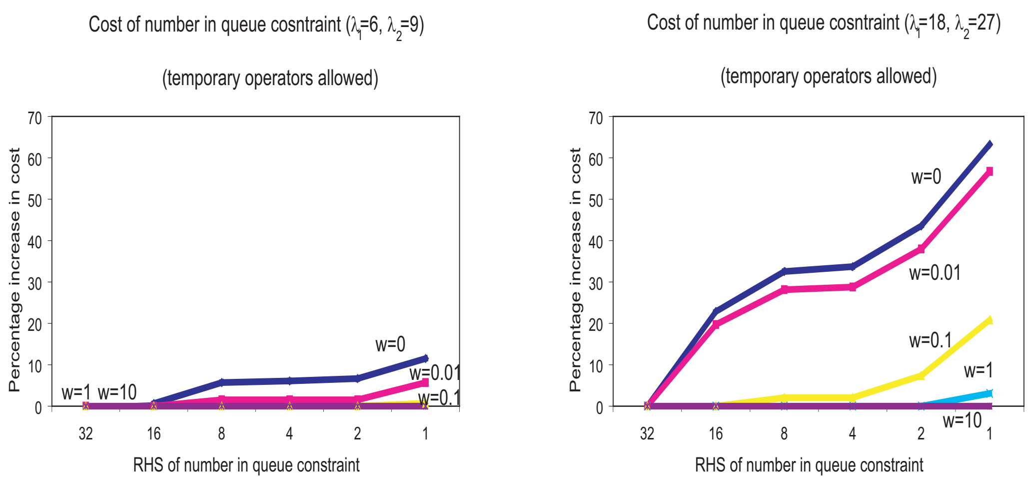

Figure 3.10: Economic cost of satisfying the delay constraint with up to five temporary operators.
$\mu_1 = \mu_2 = 2, \alpha(i) = 0.5, \beta(i) = 0.5, Nt = 50, d = 0, c_1 = 1, c_2 = 0.1, c_3 = 0.4.$

3.4.4 Computation Speed

While the $(\mathcal{IP})$ formulation in Section 3.3 yields the optimal non-randomized policy, in the worst case scenario the solution algorithm may visit every node of the solution tree, and the solution time may, in theory, be worse than enumeration. Hence, we carry out experiments for the CDOS problem comparing the computation time of the MBEDC algorithm against enumeration, as enumeration is the current fastest exact solution method for the CDOS problem. (We also investigated other LP and MIP formulations, see chapter 2.) We vary the values of the different input parameters to the CDOS problem as detailed in Table 3.1.

Parameter	Description	Number of Possible Values	Values
λ_1	Low arrival-rate	3	6, 12, 18
λ_2	High arrival-rate	3	9, 18, 27
μ_1	Permanent operator service rate	3	0.5, 1, 2
μ_2	Temporary operator service rate	3	0.5, 1, 2
c_1	Time-average permanent operator cost	4	0.01, 0.1, 1, 10
c_2	Time-average temporary operator stand-by cost	4	0.001, 0.01, 0.1, 1
c_3	Cost of temporary operator providing service	4	0.004, 0.04, 0.4, 2
d	Cost of delay per customer	4	0.01, 0.1, 1, 10
w	Cost per customer waiting	2	0, 0.01
P	Right hand side of no delay constraint	8	0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9
N	Maximum number of jobs in system	3	50, 100, 150
x	Number of permanent operators	40	4, 5, 6, . . . , 43
y	Number of temporary operators	40	1, 2, 3, . . . , 40
n_1	Lower arrival-rate process threshold	100	4, 5, 6, . . . , 103
n_2	Higher arrival-rate process threshold	100	4, 5, 6, . . . , 103
α	Rate of transition from low to high	3	0.05, 0.5, 5
β	Rate of transition from high to low	3	0.05, 0.5, 5

Table 3.1: Parameter settings for the CDOS problem.

Figure 3.11 gives the ratio of the time taken by enumeration to the time taken by MBEDC algorithm; obviously for the MBEDC algorithm to be efficient, this ratio must be greater than 1.

In all our experiments this ratio is always greater than 1, with minimum value 1.125, maximum value 200, and an average value of 10.5. We also observe from Figure 3.11 that the ratio is greatest for large problem sizes (as choices of n_1 and n_2 increase and also as N increases). This is an important result. As an example, for 40 choices each for x, y, n_1 and n_2 (2560000 combinations) with $N = 150$, the MBEDC algorithm required 46 minutes on average to solve the problem, while we estimate that enumeration would require 28 hours, resulting in a ratio of 36.5. (We did not perform 2560000 enumerations. We performed 1000 sample enumerations to obtain the average time taken per enumeration and then scaled. The variance in the 1000 samples was negligible.)

While the MBEDC solution time for the CDOS problem is not in seconds, this algorithm can still be used in practice. This is because the CDOS problem is an off-line problem, i.e., the call center manager does not solve this problem in real-time. If a heuristic approach with shorter computation times is preferred, the MBEDC algorithm can still be applied to benchmark the heuristic against optimal solutions. Finally, we note that the MBEDC algorithm using the LP relaxation of $(\mathcal{IP})$ is faster than enumeration for the CDOS problem without constraints, but slower than the existing LP formulation shown in section 3.2. This is not surprising because the LP relaxation of $(\mathcal{IP})$ has additional redundant constraints (3.13).

3.5 Extensions of the CDOS Model

In this section we consider various extensions of our model for the CDOS problem.

For the arrival process, our model extends to a general l-state MMPP process ($l > 2$). It also extends to the more general BMAP arrival processes. Under these assumptions, the CDOS problems can still be represented as a MDP with constraints and hence, can still be solved using the MBEDC algorithm; but under these models the corresponding Markov chains become more complex and hence the problem size will increase. However, based on our experiments, the MBEDC algorithm's relative (to enumeration) performance appears to improve as problem size increases. Hence, we believe that the MBEDC algorithm will still be comparably efficient. The model also extends to phase-type service distributions where a similar argument holds.

The model also easily extends to an infinite state space (infinite queue length) as long as it is infinite in only one dimension. This is because the resulting Markov chain repeats after some finite state, so it can be analyzed using the matrix analytic technique (Neuts, 1981). Intuitively, the infinite state space Markov chain can be transformed into a equivalent finite state space Markov chain. Computational results for the infinite state space problem are identical to the

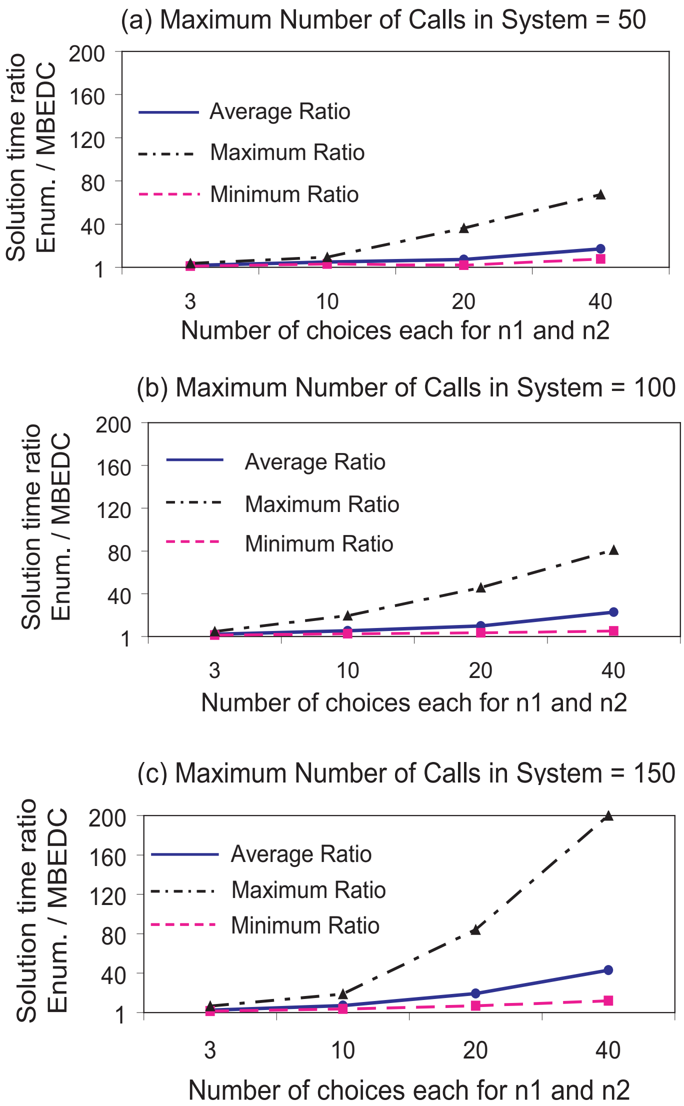

Figure 3.11: Computation time comparison.

results in Figure 3.11.

In practice, customers may not wait very long before abandoning their call, especially if the call is not free. Thus, the call center manager may be interested in modeling this phenomenon and/or introducing either a service level target or an opportunity cost, or both, on the proportion of customers abandoning their call. One way to model customer abandonment is to include exponential transition rates corresponding to customers giving up their call for each state; these rates may be increasing in the number of jobs in system. For a finite state system, this still yields a constrained MDP solvable by the MBEDC algorithm. For an infinite state system, an assumption that these rates have a maximum upper bound is required to ensure that the matrix analytic technique can be easily applied. Given this assumption, or a finite state system, it is easy to include an opportunity cost or a service level target, or both, on the proportion of customers abandoning their call.

Finally, one extreme scenario that may occur with threshold policies is that the number of customers varies quickly over time, and thus the thresholds are crossed repeatedly over a short time interval resulting in starting and stopping the temporary operators several times. This issue can be addressed in the model in two ways: (i) Assign a fixed cost that is incurred every time the temporary operators start providing service, or (ii) choose the lower threshold values $m_1 = m_2 = 0$ (instead of x), i.e., once the temporary operators start providing service they continue to be available until there are no jobs in the system.

3.6 Short Summary

The CDOS problem is a very relevant problem not only in call centers but in many service industries such as flexible Internet servers (Akamai Technologies, www.akamai.com), restaurants, banks, grocery stores, etc. We model the CDOS problem as a constrained Markov decision process (MDP) and seek the optimal non-randomized policy. We provide a novel exact and efficient solution method, the MBEDC algorithm, yielding a Mixed Integer Program formulation; the computation times of this algorithm for sample problems are lower than enumeration by up to a factor of 200, and by a factor of 10 on average. Using our algorithm, we quickly solve diverse instances of the CDOS problem, generating managerial insights into the effects of temporary operators and service level constraints. A more detailed summary is provided in section 7.2.

In the next chapter, we look at two other applications of constrained Markov decision process from the literature. We obtain the non-randomized optimal policy for these applications using the MBEDC algorithm.

Chapter 4

Other Applications of Constrained Markov Decision Process

In this chapter, we apply the MBEDC algorithm to two other problems from the literature. The first problem involves developing a framework for comparing threshold based allocation policies with different numbers of thresholds to the general optimal policy for a multiserver queuing system. While multiple threshold based policies have been proposed and analyzed in the literature, there is no unifying framework that answers the question: How many thresholds are actually required in the real world to obtain near optimal (with respect to some given metric) performance? Any threshold based allocation policy for a multiserver queuing system corresponds to a constrained Markov decision process, thus we employ the MBEDC algorithm to obtain the optimal threshold values. We also obtain the optimal state dependent policy from the solution of an unconstrained Markov decision process. In particular, we analyze the three-threshold adaptive dual threshold (ADT) policy proposed in Osogami et al. (2005) and compare its performance to the general state dependent optimal policy with respect to the weighted (over all customers) mean response time. The framework in section 4.1 extends to the comparison of performances of any two threshold based policies or the comparison of performances of one threshold based policy and the general optimal policy for a multiserver queuing system.

The second problem is a multi-item two location spare parts problem with lateral transshipments and waiting time constraints. The objective is to find the optimal non-randomized policy so as to minimize total inventory costs subject to the waiting time constraints at each location. Wong et al. (2006) study different heuristics to solve this problem and compare the gap between the heuristic solutions and a lower bound obtained by lagrangian relaxation. We solve this problem by using the MBEDC algorithm in section 4.2, providing a tool for comparing heuristic solutions to the optimal solutions, rather than a lower bound.

4.1 Framework for Comparing Threshold Based Policies in Multiserver Systems

For many organizations in the service industry, the problem of allocation of multiple servers (resource) to different queues (customer types) so as to optimize some service metric is of critical interest. Such multiserver systems with multiple queues are commonly used in call centers, repair facilities, etc. Many authors (Osogami et al., 2005; Squillante et al., 2001; Williams, 2000; Meyn, 2001; Ahn et al., 2004) have proposed and analyzed (multiple) threshold based policies for allocating the servers to queues, with the number of thresholds ranging from one to infinity. However, to the best of our knowledge, there is no work in the literature that answers the question: How many thresholds are actually required in practice for allocation of multiple servers to multiple queues to obtain near optimal (with respect to some given metric) performance?

Focus. Given a multiserver system with multiple queues, we develop a general framework for comparing any threshold based allocation policy to the optimal allocation policy with respect to the weighted (over all customers) mean response time metric. As an example of our technique, we compare a three-threshold policy, the ADT policy (Osogami et al., 2005), with the (state-dependent) optimal policy with respect to the weighted mean response time metric. Complicating this problem is the fact that we are interested only in non-randomized allocation policies (non-randomized policies have been defined in chapter 3) because of practical considerations: Randomized policies are typically difficult to implement in practice and obtaining the randomization probabilities itself is known to be difficult (Shwartz and Makowski, 1990; Nain and Ross, 1986).

Research questions and contributions. We answer the following research questions for this problem:

- Is there a general framework for the comparison of any threshold based allocation policy to the optimal allocation policy with respect to any service metric?

We develop a framework for the comparison of the three-threshold ADT policy to the optimal policy with respect to weighted mean response time metric and show how to extend our approach to more general comparisons. Section 4.1.1 describes the ADT and optimal policies in our problem setting. The mathematical programming formulations for these two policies (Linear Program and Mixed Integer Program, respectively) are provided in section 4.1.2. Note that the optimal policy can be obtained from the solution to a Markov decision process for which three solution techniques exist: Policy iteration, value iteration, and linear programming. However, the ADT policy, and in fact any finite-number-of-thresholds policy, results in a constrained Markov decision process. There is no existing solution technique to obtain the optimal non-randomized solution to a constrained Markov decision process. However, we can now use the MBEDC algorithm developed in this thesis to solve such problems. While we consider the ADT policy in this thesis, the MBEDC algorithm can be applied to obtain the optimal threshold policy for any number of finite thresholds, resulting in a general framework for comparing threshold policies.

Section 4.1.3 presents computational results for the comparison of the ADT policy with the optimal policy with respect to the weighted mean response time for different values of the load (ratio of arrival rate to service rate) at the servers. We show that the percentage decrease in the weighted mean response time resulting from increasing the number of thresholds from three (optimal ADT policy) to infinity (optimal state-dependent policy) is only 0-3% in our experiments. Thus, for the weighted mean response time service metric, having one to three thresholds provides good (within 3% of the optimal) performance in terms of the objective function, and these may be easier to implement compared to a state-dependent policy. This is another strong argument in the favor of threshold-based policies.

Related Literature. In Osogami et al. (2005), the authors compare the optimal (three-threshold) ADT policy to the optimal (one-threshold) T1 policy with respect to the weighted mean response time metric. To the best of our knowledge, there is no work that compares optimal threshold based policies to the optimal (state-dependent) policy with respect to a service metric.

4.1.1 Description of ADT and Optimal Policies

First, we describe our problem setting of a multiserver queuing system such as those commonly used in modeling call centers, repair facilities, etc. This setting is similar to the one studied in Osogami et al. (2005). Specifically, we study a multiserver model that consists of two servers and two queues (Beneficiary-Donor) model, as shown in Figure 4.1. The jobs arrive at queue 1 and queue 2 according to Poisson processes with arrival rates λ_1 and λ_2, respectively. Jobs have exponentially distributed service times requirements which are server dependent. Server 1 (beneficiary server) processes jobs in queue 1 (type 1 jobs) with rate μ_1, while server 2 (donor server) can process type 1 jobs with rate μ_{12}, and can process jobs in queue 2 (type 2 jobs) with rate μ_2. Note that if $\mu_{12} < \mu_1$, then the problem is easy and the optimal allocation policy in that case is the $c\mu$ rule (Cox and Smith, 1971). However, if $\mu_{12} > \mu_1$ (server 2 is a faster server overall) then the $c\mu$ rule may be unstable (Squillante et al., 2001), i.e., queue lengths may grow unboundedly. Define $\rho_1 = \lambda_1/\mu_1$, $\rho_2 = \lambda_2/\mu_2$, and $\hat{\rho}_1 = \lambda_1/(\mu_1 + \mu_{12}(1 - \rho_2))$ as the loads of queue 1 without any help from server 2, of queue 2, and of queue 1 when server 2 helps server 1, respectively. Given these input parameters, the goal is to determine when the donor server should work on its own jobs and when it should work on the beneficiary server jobs so as to minimize the overall weighted mean response time $c_1 p_1 E[R_1] + c_2 p_2 E[R_2]$, where c_i is the weight of type i jobs, $p_i = \lambda_i/(\lambda_1 + \lambda_2)$ is the fraction of type i jobs, and $E[R_i]$ is the mean response time of type i jobs, for $i = 1, 2$. The focus is on non-randomized threshold-based policies. Refer to Osogami et al. (2005) for stability conditions of the system, and for a literature review on the topic of Beneficiary-Donor models.

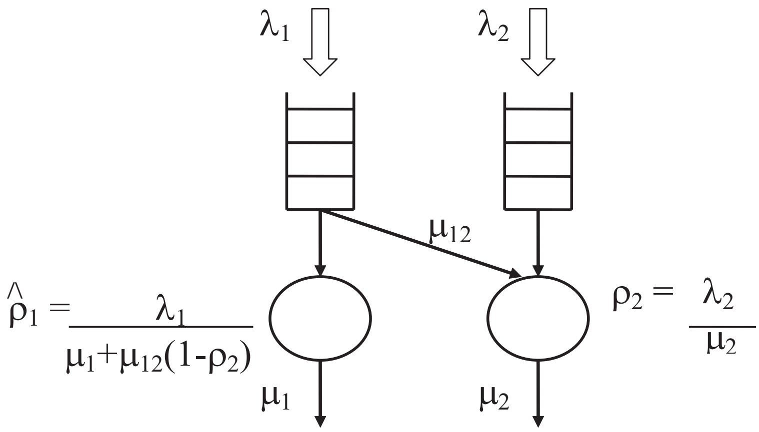

Figure 4.1: Beneficiary-Donor model.

One threshold based policy suggested for this problem in the literature is the T1 policy (Squillante et al., 2001; Williams, 2000). The T1 policy places a threshold, t_1, on queue 1, so that server 2 processes type 1 jobs only when there are at least t_1 jobs of type 1, or if queue 2 is empty. It is also known (Meyn, 2001; Ahn et al., 2004) that the T1 policy is not optimal in general. Meyn (2001) obtains, via a numerical approach, the optimal allocation policy when both queues have finite buffers. Although not proven, the optimal threshold policy appears to be a “flexible” T1 policy that allows a continuum of T1 thresholds, one for each possible length of queue 2. The optimal state-dependent policy can be obtained from the solution of a Markov decision process; the formulation is provided in Section 4.1.2.

In Osogami et al. (2005), the authors propose a class of multi-threshold allocation policies: Adaptive Dual Threshold (ADT) policies. Specifically, the ADT policy behaves like the T1 policy with parameter $t_1^{(1)}$ if the length of queue 2 is less than t_2, and otherwise like the T1 policy with parameter $t_1^{(2)}$, where $t_1^{(2)} > t_1^{(1)}$ as shown in Figure 4.2. Note that the ADT policy is equivalent to the T1 policy if $t_2 = \infty$. The authors compare the optimal ADT policy to the optimal T1 policy, and find that the mean response time obtained from the optimal T1 policy is, at worst, close to the mean response time obtained from the optimal ADT policy. This is a surprising result, since the ADT policy generalizes the T1 policy, and since the general optimal policy appears to have infinitely many thresholds. The authors conjecture that adding more thresholds (approaching the general optimal policy) will not improve mean response time appreciably compared to the ADT policy.

We test the above conjecture in this section by finding the optimal general allocation policy, and the optimal ADT policy, both by solving Markov decision processes in which we define the state of the system as the number of jobs in queue 1 and the number of jobs in queue 2. In each state, there are two action choices: (i) Ask server 2 to process type 1 jobs, (ii) ask server 2 to process type 2 jobs. In the general allocation policy case there are no probabilistic constraints, so solving an LP yields the optimal policy, which exhibits a structure similar to that found by (Meyn, 2001). On the other hand, the optimal ADT policy is obtained by solving a constrained MDP. This is because the requirements that the parameters, $t_1^{(1)}$, t_2, and $t_1^{(2)}$, be fixed for all states of the system are constraints on the limiting probability values of the MDP. Thus, we implement the MBEDC algorithm to obtain the optimal ADT policy. Next, we provide the formulations for finding the optimal state-dependent policy and the optimal ADT policy.

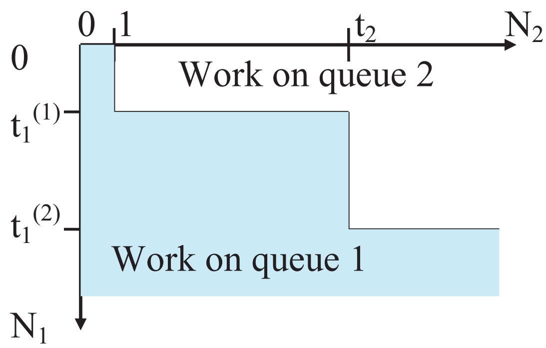

Figure 4.2: Queue on which server 2 works as a function of number of jobs in queue 1 (N_1) and number of jobs in queue 2 (N_2) under the ADT policy.

4.1.2 Formulations for the Optimal and ADT Policies

Denote by state $s = (i, j) \in S$, that the number of jobs in system of types 1 and 2 are i and j, respectively. (Number of jobs in system is the total of the number of jobs in queue plus the number of jobs being served.) Let N_k denote the maximum number of jobs in system for server k, $k = 1, 2$; thus, we assume a finite queue. In the case of infinite length queues, an approximation scheme called dimensionality reduction proposed in Harchol-Balter et al. (2003) can be used, or truncation can be applied to get the mean response time and the queue length distribution. The matrix analytic method developed by (Neuts, 1981) can most easily be applied to problems with Markov chains that are infinite in one dimension; hence, it cannot easily be applied to this problem as is. In this section, we model the problem of finding the optimal state-dependent policy as a Markov decision process, and present the linear programming formulation for obtaining the optimal policy. We model the problem of finding the optimal ADT policy as a constrained Markov decision process. We use the MBEDC algorithm developed in this thesis to obtain a mixed integer programming formulation which yields the optimal non-randomized ADT policy. The only other method to obtain the exact, non-randomized, optimal ADT policy is enumeration. Before we present the formulations, some additional notation is introduced below.

We define $\pi_{s,a}$ to be the limiting probability that the MDP is in state s and the call center manager chooses action a. Define A_s, the set of possible action choices in state s as

$$A_s = \begin{cases} 0 & : \text{ Donor server works on Donor server's jobs} \\ 1 & : \text{ Donor server works on Beneficiary server's jobs} \\ 2 & : \text{ Donor server works on Donor server's jobs} \\ 3 & : \text{ Donor server works on Beneficiary server's jobs} \end{cases}$$

While action indices 0 and 2, and 1 and 3, result in the same action, respectively, we need to define all four indices because of the ADT policy. This will be explained in detail later when we discuss the formulation for the ADT policy. Action choices 0, 1, 2 and 3 correspond to the cases that no thresholds have yet been crossed, threshold $t_1^{(1)}$ has been crossed but not t_2, threshold t_2 has been crossed but not $t_1^{(2)}$, and threshold $t_1^{(2)}$ has been crossed. It appears to not be possible to use the MBEDC algorithm (developed in this thesis to solve CMDPs) with only two action choices (0 and 1) to obtain the optimal ADT policy as this results in an infeasible integer programming formulation. On the contrary, to obtain the general optimal policy, we only need an unconstrained Markov decision process, and hence, two action choices (0 and 1) suffice.

Define $\Gamma_{s,a}$ is the total transition rate out of state s if action a is selected, and $\gamma_{j|s,a}$ as the rate of transition to state j if action a is selected in state s. For example, consider the state in which the number of jobs in system are 3 and 2 for types 1 and 2, respectively, i.e.,

$s = (3,2)$. Suppose that the lower threshold for system 1 is crossed ($t_1^{(1)} \leq 3$) but the threshold for system 2 has not been reached ($t_2 \geq 3$), i.e., $a_{(3,2)} = 1$. Then the total rate out of this state is $\Gamma_{(3,2),1} = \lambda_1 + \lambda_2 + \mu_1 + \mu_{12}$. Next consider the state in which the number of jobs in system is 2 for both servers, i.e., $s = (2,2)$. Suppose that the threshold for server 2 is crossed ($t_2 \leq 2$) but the higher threshold for server 1 has not been reached ($t_1^{(2)} \geq 3$), i.e., $a_{(2,2)} = 2$. The rate of transition from this state to state (2,1) is $\gamma_{(2,1)|(2,2),2} = \mu_2$. The LP formulation to obtain the general optimal threshold policy is given below. It yields the optimal non-randomized solution since there are no probabilistic constraints.

Formulation for the optimal general threshold policy

Note that for the general threshold policy, A_s consists of only two action choices for all states s: 0 and 1. Choice 0 corresponds to the donor server working on its own jobs and choice 1 corresponds to the donor server working on the beneficiary server's jobs. Let w_k be the objective weight associated with the mean response time of class k. Since we do not put any restrictions (constraints) on the limiting probability values other than the modified balance equations, the decision maker is free to choose either of the actions in any state.

$$\text{MIN} \quad w_1 \left(\sum_{i=1}^{N_1} \sum_{j=0}^{N_2} \sum_{a \in A_s} \pi_{s,a} \; i \right) \; + \; w_2 \left(\sum_{j=1}^{N_2} \sum_{i=0}^{N_1} \sum_{a \in A_s} \pi_{s,a} \; j \right)$$

Subject to

$$\sum_{a \in A_j} \Gamma_{j,a} \, \pi_{j,a} - \sum_{s \in S} \sum_{a \in A_s} \gamma_{j|s,a} \, \pi_{s,a} = 0 \quad \text{for all } j \in S$$

$$\sum_{s \in S} \sum_{a \in A_s} \pi_{s,a} = 1$$

$$\pi_{s,a} \geq 0 \quad \text{for all } (s,a) \text{ such that } s \in S, \; a \in A_s$$

To obtain the optimal ADT threshold policy using the MBEDC algorithm, we introduce more notation. Let b_i, d_j, and o_i denote binary variables for all $i = 1, \ldots, N_1$ and for all $j = 1, \ldots, N_2$ that take the value one when $t_1^{(1)} = i$, $t_2 = j$, and $t_1^{(2)} = i$, respectively, and zero otherwise. We allow the decision maker to choose between including and not including any one of these thresholds by letting b_{N_1+1}, d_{N_2+1} and o_{N_1+1} be binary variables that take value one if these thresholds are not required and zero otherwise. (We choose b, d, and o to represent beneficiary's first threshold, donor's threshold, and finally the overall system threshold also known as the beneficiary's second threshold.) The formulation for the optimal ADT policy is presented below.

Formulation for the optimal ADT policy

$$\text{MIN} \;\; w_1 \left(\sum_{i=1}^{N_1} \sum_{j=0}^{N_2} \sum_{a \in A_s} \pi_{s,a}\; i \right) \; + \; w_2 \left(\sum_{j=1}^{N_2} \sum_{i=0}^{N_1} \sum_{a \in A_s} \pi_{s,a}\; j \right)$$

Subject to

$$\sum_{a \in A_j} \Gamma_{j,a}\; \pi_{j,a} - \sum_{s \in S} \sum_{a \in A_s} \gamma_{j|s,a}\; \pi_{s,a} = 0 \quad \text{for all } j \in S$$

$$\sum_{s \in S} \sum_{a \in A_s} \pi_{s,a} = 1$$

$$\pi_{s,a} \geq 0 \quad \text{for all } (s,a) \text{ such that } s \in S,\; a \in A_s$$

$$\sum_{i=2}^{N_1+1} b_i = 1, b_i \in \{0,1\} \;\forall\; i = 2, \ldots, N_1+1$$

$$\sum_{l=2}^{i-1} \sum_{j=1}^{N_2} \pi_{(l,j),1} + \sum_{l=i}^{N_1} \sum_{j=1}^{N_2} \pi_{(l,j),0} \leq 1 - b_i,\; i = 2, \ldots, N_1+1$$

$$\sum_{j=1}^{N_2+1} d_i = 1, d_i \in \{0,1\} \;\forall\; j = 1, \ldots, N_2+1$$

$$\sum_{i=2}^{N_1} \sum_{k=1}^{j-1} \pi_{(i,k),2} + \sum_{i=2}^{N_1} \sum_{k=j}^{N_2} \left(\pi_{(i,k),0} + \pi_{(i,k),1} \right) \leq 1 - d_j,\; j = 1, \ldots, N_2+1$$

$$\sum_{i=2}^{N_1+1} o_i = 1, o_i \in \{0,1\} \;\forall\; i = 2, \ldots, N_1+1$$

$$\sum_{l=2}^{i-1} \sum_{j=1}^{N_2} \pi_{(l,j),3} + \sum_{l=i}^{N_1} \sum_{j=1}^{N_2} \left(\pi_{(l,j),0} + \pi_{(l,j),1} + \pi_{(l,j),2} \right) \leq 1 - o_i,\; i = 2, \ldots, N_1+1$$

We explain the objective first. The first and the second terms of the objective give the weighted mean response time of job types 1 and 2, respectively (using Little's Law). Hence, the objective gives the total weighted mean response time. The first three constraints are the same as those in the LP formulation above, and yield the limiting probability values. The remaining constraints ensure that for each decision parameter (b, d and o corresponding to $t_1^{(1)}$, t_2 and $t_1^{(2)}$, respectively) exactly one value is assigned, and the linking constraints ensure that the limiting probability values relate to the corresponding binary variables. These constraints are constructed using the sets Q_i^k defined for this problem. Refer to section 3.3 for a discussion of these sets; Q_i^k contains the state-action pairs that cannot be selected if action k is selected in state i, and vice versa. For example for b_6, the set $Q_6^{(b)}$ is given by,

$$Q_6^{(b)} = \left\{ \begin{array}{l} \pi_{(2,1)1}, \pi_{(2,2)1}, \ldots, \pi_{(2,N_2)1} \\ \pi_{(3,1)1}, \pi_{(3,2)1}, \ldots, \pi_{(3,N_2)1} \\ \pi_{(4,1)1}, \pi_{(4,2)1}, \ldots, \pi_{(4,N_2)1} \\ \pi_{(5,1)1}, \pi_{(5,2)1}, \ldots, \pi_{(5,N_2)1} \\ \pi_{(6,1)0}, \pi_{(6,2)0}, \ldots, \pi_{(6,N_2)0} \\ \pi_{(7,1)0}, \pi_{(7,2)0}, \ldots, \pi_{(7,N_2)0} \\ \vdots \\ \pi_{(N_1,1)0}, \pi_{(N_1,2)0}, \ldots, \pi_{(N_1,N_2)0} \end{array} \right\}$$

The process for developing the mixed integer programming formulation for any finite number of thresholds policy is similar to that for the ADT policy. The key is to define constraints that ensure that for each decision parameter (corresponding to each threshold), the corresponding binary variables sum to one, i.e., the decision parameter has exactly one value. In addition, constraints that link the limiting probability variables to the corresponding binary variables must be included. These constraints are developed by appropriate defining the sets Q_i^k as discussed in section 3.3.

4.1.3 Numerical Experiments

We present below in Figure 4.3 a reproduction of Figure 10 from Osogami et al. (2005) in which the authors show that the ADT policy does not significantly improve over the T1 policy with respect to the weighted mean response time even though the ADT policy generalizes the T1 policy. However, they do not find the globally optimal threshold values for the ADT policy. Since the search space of the threshold values for the ADT policy is large, they find locally optimal threshold values, which are found to be optimal within a search space of ± 5 for each threshold using simulation. They conjecture that adding more thresholds (approaching the optimal flexible T1 policy) will not improve mean response time appreciably.

We test this conjecture using the same data as in Figure 10 of Osogami et al. (2005). The results of our experiments are shown in Figure 4.4. We find the same ADT solutions as Osogami et al. (2005); their method finds globally optimal ADT parameters. Our experiments also show that their conjecture is indeed true for these examples; the maximum improvement is 2.35%. The computation time in seconds for the experiments in Figure 4.4 are shown in Table 4.1. We assume a maximum queue length of 50 for both queues. With these values for the maximum queue length, we see a truncation effect for queue 2 in Figure 4.4 and Table 4.1 when $\rho_2 = 0.8$.

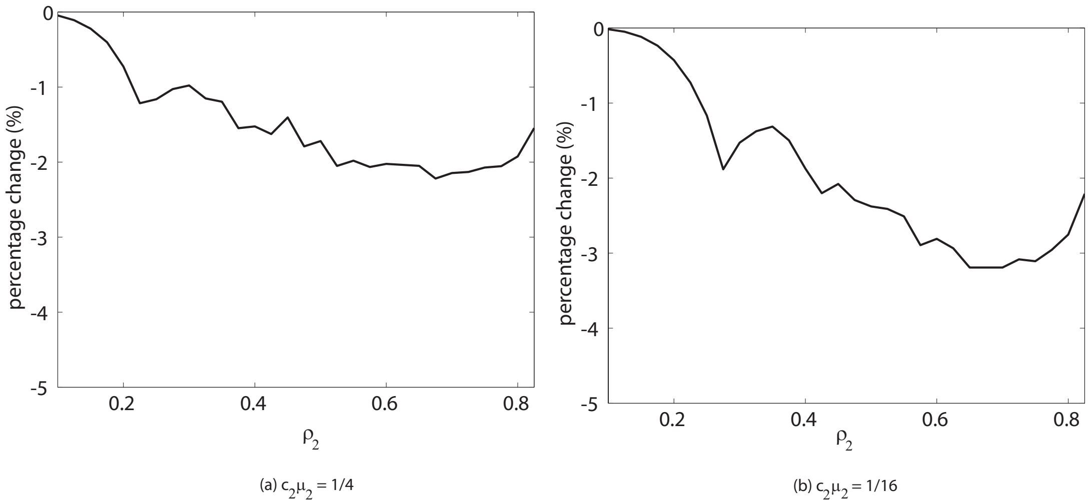

Figure 4.3: The percentage change in the mean response time of the (locally) optimized ADT policy over the optimized T1 policy at each given load, as a function of ρ_2. A negative percentage indicates the improvement of ADT over T1. Here, $c_1 = c_2 = 1$, $c_1\mu_1 = c_1\mu_{12} = 1$, and $\rho_1 = 1.15$ are fixed.

As a result of truncation, the probability mass in the tail of system 2 is relatively high, and the expected number in system and hence the mean response time is underestimated as the number in system cannot be more than 50, which would not be the case in system 2 if it had an infinite buffer. However, increasing the maximum queue length of both queues greatly increases the solution time of the problem. Hence, for $\rho_2 = 0.8$, we choose the maximum queue lengths of 15 and 150 for queues 1 and 2, respectively. These values are selected after running some trial experiments to determine suitable maximum queue length values with the tradeoff between reducing the truncation effect and not increasing the complexity of the problem greatly. We indicate this difference in input parameters for $\rho_2 = 0.8$, by putting an asterix $(*)$ next to 0.8 in Figure 4.4 and Table 4.1. One can also truncate only the queue for server 1 while considering an infinite buffer for server 2. Neuts' matrix analytic method can now be used since the resulting Markov chain is infinite in only one dimension.

We observe that the percentage decrease in the weighted response time from changing to an optimal flexible T1 policy from the ADT policy is very small at low values of ρ_2, the load at queue 2. This percentage benefit increases with ρ_2, the benefit of adding more thresholds is the highest for high values of ρ_2. We see a similar distinction for the computation times of the optimal ADT policy. The optimal flexible T1 policy is obtained fairly quickly since it involves solving a Linear Program. The computation times of the optimal ADT policy are fairly low for small values of ρ_2 and increases with ρ_2. (At $\rho_2 = 0.8$, the maximum queue lengths of the two servers are different compared to all other values of ρ_2 as mentioned earlier. Hence, its

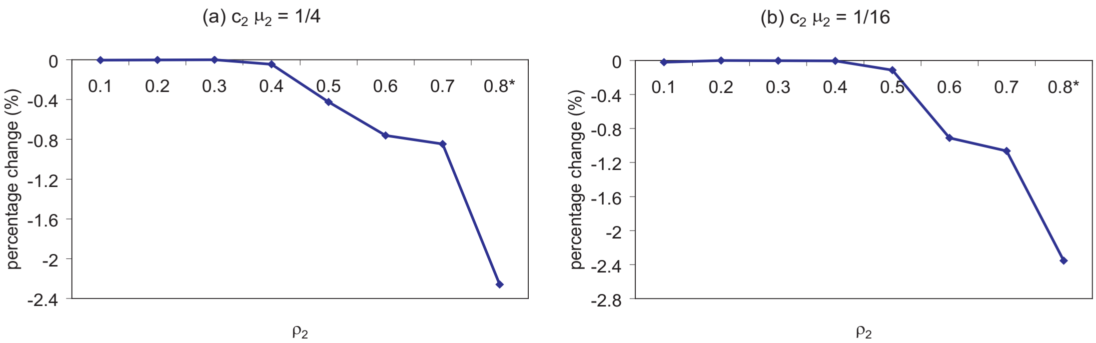

Figure 4.4: The percentage change in the mean response time of the optimized threshold policy over the optimized ADT policy at each given load, as a function of ρ_2. Here, $c_1 = c_2 = 1$, $c_1\mu_1 = c_1\mu_{12} = 1$, and $\rho_1 = 1.15$ are fixed.

computation time is not representative.)

ρ_2	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8***
				Figure 4.4(a)				
General	19.09	12.01	21.36	7.61	14.6	22.4	27.8	2.39
ADT	15.06	11.21	9.7	30.66	80300	94241	124330	897.68
				Figure 4.4(b)				
General	3.26	6.3	19.45	19.65	22.01	20.48	26.22	2.45
ADT	15.91	12.78	10.95	10.88	157	56671	122330	12237.21

Table 4.1: Computation time in seconds.

This problem demonstrates another application of the MBEDC algorithm. Without the MBEDC algorithm, it would not have been possible to obtain the optimal non-randomized ADT threshold policy more efficiently than enumeration. Next, we discuss the multi-item spare parts problem with applications in inventory management.

4.2 The Multi-Item Spare Parts Problem

The second problem we study from the literature is the multi-item two echelon spare parts system with lateral transshipments and waiting time constraints (Wong et al., 2006). The system under consideration is shown in Figure 4.5 and consists of a central warehouse and two local warehouses; there is a target for the aggregate (over all items) mean waiting time per local warehouse. Such a setting is commonly used by many manufacturers of "high-tech" products (airplanes, complex machinery, medical equipment, mainframes, etc.) to support their after-sales services, as downtime of machines at their customers can be very costly. The problem faced by the manufacturer is to determine the optimal base-stock levels of spare parts (one base-stock level for each item) at the two local warehouses so that target aggregate mean waiting times are met against minimal inventory holding costs (since many parts are very expensive). A base-stock policy is a decision rule that decides a base-stock level S such that if inventory position (inventory on hand plus on order) at any time is at or above S no order is placed, and if the inventory at any time is strictly below S then an order is placed to bring the inventory position to S. If an item is demanded at one local warehouse which does not have it on hand, the item can be shipped from the other local warehouse provided it has the item on stock. This is called lateral transshipment (also referred to as inventory pooling) and represents an effective strategy to reduce downtime at the customer and at the same time control total system inventory costs for the manufacturer. The central warehouse is assumed to be an ample server and emergency shipments from this warehouse are also allowed in the case that both local warehouses do not have the demanded item on hand.

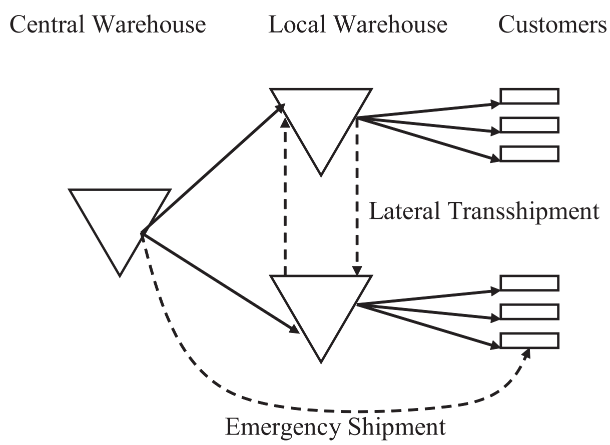

Figure 4.5: Supply chain of the spare-parts system.

Refer to Wong et al. (2006) for more details and a literature review for this problem. We use the same assumptions as made in that paper and discuss them below. The items are indexed by $i = 1, 2, \ldots, I$ and the local warehouses are indexed by $j = 1, 2$. (i) *Failures occur according to Poisson processes with constant rates.* For expensive technical systems, failure rates of parts are low in general and thus downtimes do not occur very often. It is quite common in the literature on spare parts inventory models (see for example Sherbrooke, 2004) to assume constant failure rates according to Poisson processes for each spare part. The demand rate of components of item i at warehouse j is given by m_{ij}. Further, $M_j = \sum_{i=1}^{I} m_{ij}$ denotes the total demand rate at warehouse j. (ii) *All parts are repairable and there is no condemnation*, i.e., there is no procurement of new items to replace those items that can no longer be repaired. (iii) *Lateral transshipments are faster and cheaper than emergency supplies.* Typically, the two local warehouses are closer to the clients location than the central warehouse. Therefore, a lateral transshipment is always preferred above an emergency shipment. (iv) *The repair lead times are exponential with rate μ_i for item i.* This assumption facilitates the analysis using a Markov decision process. (v) *Complete pooling is applied.* Thus, the two local warehouses have to agree to a lateral transshipment if they have the item on stock. In case the two warehouses do not want to share their last parts, one may introduce threshold parameters such that the warehouse agrees to a lateral transshipment only if its own stock level for that item is above that threshold. (vi) *Inventory holding costs dominate shipping costs.* This assumption is justified in practice since the parts we consider are very expensive. Thus, the focus is on minimizing total inventory costs.

Let S_j^i be the base-stock level of item i at warehouse j. If a demand for item i at warehouse j is met from stock the downtime is assumed to be zero without loss of generality. If a lateral transshipment from the other warehouse is required (this happens when the original location does not have the demanded item on stock), the mean downtime is assumed to be t_1. If the second warehouse also does not have item on stock, then an emergency shipment is made from the central warehouse, and the mean downtime is t_2 ($> t_1$). At warehouse j, there is a target level, W_j, for the maximum value of the weighted average waiting time (downtime) over all items. Let the holding cost for item i be c^i. We ignore shipping costs as these are small compared to the cost of the items which can be very expensive; our model easily extends to the case when these costs are included. The objective is to find a base-stock policy (S_1, S_2), where S_1 and S_2 are vectors of base-stock levels of all items at local warehouses 1 and 2, respectively, under which the total inventory costs are minimized subject to the waiting time constraints. The state of the art technique to obtain the optimal non-randomized policy is a greedy heuristic developed by Wong et al. (2006). They compare the performance of this heuristic with a lower bound obtained from lagrangian relaxation, but not the optimal policy. The performance of this heuristic is good and it results in an average gap of 1-5% on their sample problems. However, there has been no study in the literature to obtain the exact optimal non-randomized solution for this problem.

The Markov chain associated with the choice of policy (S_1^i, S_2^i) = (2,1) for item i is shown in Figure 4.6. Define π^i as the steady-state probability vector, and $\pi_{k,l}^i$ as the steady-state probability of being in state (k, l), $0 \le k \le S_1^i$, $0 \le l \le S_2^i$ for item i, where k and l are the number of units of item i on hand at local warehouses 1 and 2, respectively. Thus, the weighted average waiting time constraint at warehouse j is modeled as $\sum_{i=1}^{I} m_{ij}(\alpha_{ij} t_1 + \theta_i t_2) \le M_j W_j$, $j = 1, 2$, where $\alpha_{i1} = \sum_{l=1}^{S_2^i} \pi_{0,l}$, $\alpha_{i2} = \sum_{k=1}^{S_1^i} \pi_{k,0}$ and $\theta_i = \pi_{0,0}^i$. In words, α_{ij} represents the probability that a lateral transshipment is required for item i at location j, and θ_i represents the probability that an emergency shipment is required for item i (it does not depend on the location). Thus the left hand side of the above constraint gives the expected downtime for an item i weighted by the proportion of demand at location j corresponding to item i (m_{ij}/M_j). Finally, summing over all items gives the total weighted expected downtime at location j which must be less than or equal to the target level W_j. This problem is then a combination of Markov decision processes (one for each item) with probabilistic constraints linking the MDPs as they are weighted over all the Markov decision processes. We utilize the MBEDC algorithm to obtain the exact optimal solution. Since the greedy heuristic (Wong et al., 2006) is very fast and has a good performance, our aim in finding the optimal solution is to provide a tool to enable the precise quantification of the performance of any heuristic method compared to the optimal solution. This problem is more complicated than the CDOS problem studied in Chapter 3 since it involves a weighted service level constraint over different items and not a single-item problem with a service level constraint (the multi-item version would correspond to a weighted service level constraint over multiple call centers).

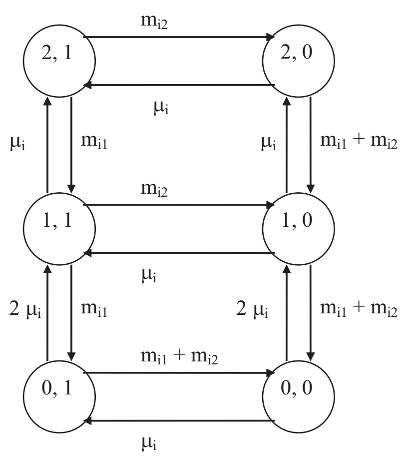

Figure 4.6: Markov chain for item i with (S_{i1}, S_{i2}) = (2, 1).

While the MBEDC algorithm is efficient compared to enumeration, it still cannot avoid the curse of dimensionality; indeed for this problem, the curse of dimensionality becomes an issue. The number of enumerations for I items with $|S|$ choices for the base-stock levels at each location is $|S|^{2I}$ which becomes large very quickly. Nevertheless, the MBEDC algorithm is still the only exact efficient solution method for this problem, and can solve problems of moderate size that heretofore were intractable.

Next, we show the formulation obtained from applying the MBEDC algorithm to this problem. For item i, let w_{ij}^u be a binary variable that indicates whether the base-stock level for item i at location j is u or not for i = 1, ..., I, j = 1, 2, and u = 0, ..., $Smax_j^i$, where $Smax_j^i$ is a known upper bound on the optimal base-stock level of item i at location j. Let the holding cost of item i be c^i. Denote by $\pi^i_{(k,l)(u,v)}$ the limiting probability that the system is in state $(k, l) = s$, and that the base-stock policy at the two locations is given by (u, v) for item i. Let $s \leq (u, v)$ imply that $k \leq u$ and $l \leq v$. Finally, $\Gamma^i_{s,(u,v)}$ is the total transition rate out of state s if action (u, v) is selected, and $\gamma^i_{j|s,(u,v)}$ is the rate of transition to state j if action (u, v) is selected in state s for item i. First, we explain the objective. The objective gives the total holding cost over all items. We include inventory in the pipeline to calculate the total holding cost and hence the objective can be expressed completely in terms of the binary variables without using the continuous limiting probability variables. We can model the case when this assumption is relaxed.

Formulation for the spare-parts problem.

$$\text{MIN} \sum_{i=1}^{I} \sum_{j=1}^{2} \sum_{u=0}^{Smax_j^i} c^i \; u \; w_{ij}^u$$

Subject to

$$\sum_{u=0}^{Smax_1^i} \sum_{v=0}^{Smax_2^i} \Gamma^i_{j,(u,v)} \; \pi^i_{j,(u,v)} - \sum_{u=0}^{Smax_1^i} \sum_{v=0}^{Smax_2^i} \sum_{s \leq (u,v)} \gamma^i_{j|s,(u,v)} \; \pi^i_{s,(u,v)} = 0 \; \forall \; j, \; \forall \; i$$

$$\sum_{u=0}^{Smax_1^i} \sum_{v=0}^{Smax_2^i} \sum_{s \leq (u,v)} \pi^i_{s,(u,v)} = 1 \; \forall \; i$$

$$\pi^i_{(k,l),(u,v)} \geq 0 \quad u \in \{0, \ldots, Smax_1^i\}, v \in \{0, \ldots, Smax_2^i\}, k \in \{0, \ldots, u\}, l \in \{0, \ldots, v\}, \; \forall \; i$$

$$\sum_{u=0}^{Smax_1^i} w_{i1}^u = 1, w_{i1}^u \in \{0, 1\} \; \forall \; i$$

$$\sum_{v=0}^{Smax_2^i} w_{i2}^v = 1, w_{i2}^v \in \{0, 1\} \; \forall \; i$$

$$\sum_{i=i}^{I} \frac{m_{ij}}{M_j} \left[t_1 \left(\sum_{(u,v)} \sum_{l=0}^{v} \pi^i_{(0,l),(u,v)} \right) + t_2 \left(\sum_{(u,v)} \pi^i_{(0,0),(u,v)} \right) \right] \leq W_1$$

$$\sum_{i=i}^{I} \frac{m_{ij}}{M_j} \left[t_1 \left(\sum_{(u,v)} \sum_{k=0}^{u} \pi^i_{(k,0),(u,v)} \right) + t_2 \left(\sum_{(u,v)} \pi^i_{(0,0),(u,v)} \right) \right] \leq W_2$$

$$\sum_{u'=0}^{u-1} \sum_{v=0}^{Smax_2^i} \sum_{k=0}^{u'} \sum_{l=0}^{v} \pi^i_{(k,l)(u',v)} + \sum_{u'=u+1}^{Smax_1^i} \sum_{v=0}^{Smax_2^i} \sum_{k=0}^{u'} \sum_{l=0}^{v} \pi^i_{(k,l)(u',v)} \leq 1 - w^u_{i1} \ \forall\ i,\ \forall\ u$$

$$\sum_{u=0}^{Smax_1^i} \sum_{v'=0}^{v-1} \sum_{k=0}^{u} \sum_{l=0}^{v'} \pi^i_{(k,l)(u,v')} + \sum_{u=0}^{Smax_1^i} \sum_{v'=v+1}^{Smax_2^i} \sum_{k=0}^{u} \sum_{l=0}^{v'} \pi^i_{(k,l)(u,v')} \leq 1 - w^v_{i2} \ \forall\ i,\ \forall\ v$$

The first three system of constraints are the modified balance equations that give the limiting probability values. The next two systems of constraints require that all binary variables corresponding to the choices for the base-stock level for item i at location j sum to one, for all i and j. These constraints ensure that exactly one value is chosen as the base-stock policy for item i at location j. The next two constraints are the waiting time target level constraints at locations 1 and 2, respectively. The last two systems of constraints link the continuous limiting probability variables to the corresponding binary variables ensuring that any feasible integer solution gives a non-randomized policy.

Computational results. Below, we show the results of applying the MBEDC algorithm for ten instances of each choice of number of items. We can easily solve this problem exactly for up to seven items as shown in Table 4.2. However, the average time required for ten items is 196418 seconds or 54.5 hours, which is very high. In some cases, the solver ran out of memory since the tree size in CPLEX became too large. We do not include these instances in Table 4.2. We tried other MDP setups (specifically state $s = (i, j)$ and choice of actions indicating: (i) base-stock level not reached at either location, (ii) base-stock level reached at location 1 but not at 2, (iii) base-stock level reached at location 2 but not at 1, and (iv) base-stock level reached at both locations) for this problem as well but they were slower than the MDP presented here.

Number of items	Minimum time (s)	Average time (s)	Maximum time (s)
2	0.62	0.83	1.05
5	15.11	33.92	52.21
7	109.7	146.56	216.92
10	50.15	196418	426923

Table 4.2: Computation time in seconds for problems with up to 10 items solved to completion.

In Table 4.3, we present the percentage gap closed by the MBEDC algorithm in a maximum of six hours of computation for up to 15 items. We notice that the MBEDC algorithm finds "good" feasible integer solutions very quickly but spends a lot of time searching through the fractional solutions corresponding to the relaxed sub-problems. The reason is that the

MBEDC algorithm introduces a lot of degeneracy in the structure of the Integer Programming formulation, i.e., multiple solutions with the same objective value are possible for the relaxed sub-problems (refer to the linking constraints) when not all the binary variables are forced to be integers. CPLEX is forced to explore multiple nodes whose resulting *children* (branches) have the same objective value as the *parents* (the node from which they were obtained). Hence, the improvements in the "best node" in CPLEX are very slow, and the time required by the MBEDC algorithm to find the optimal solution becomes large quickly as the number of items increases. The progression of CPLEX for our experiments is shown in Figure 4.7. Since CPLEX does not indicate the progression in terms of time, we show it in terms of number of nodes visited. As seen from Figure 4.7, the remaining percentage gap decreases very quickly in the beginning as CPLEX finds "good" integer feasible solutions quickly, but then improvement in the best node value and hence the closing of the remaining percentage gap is very slow. Better integer feasible solutions than the current best integer feasible solution are also obtained during this slow progression but their improvement on the current best integer feasible solution is observed to be small.

Number of items	Minimum (%)	Average (%)	Maximum (%)
2	0	0	0
5	0	0	0
7	0	0	0
10	11.71	6.06	0.03
15	19.8	16.18	0.092

Table 4.3: Remaining gap after 6 hours of computation.

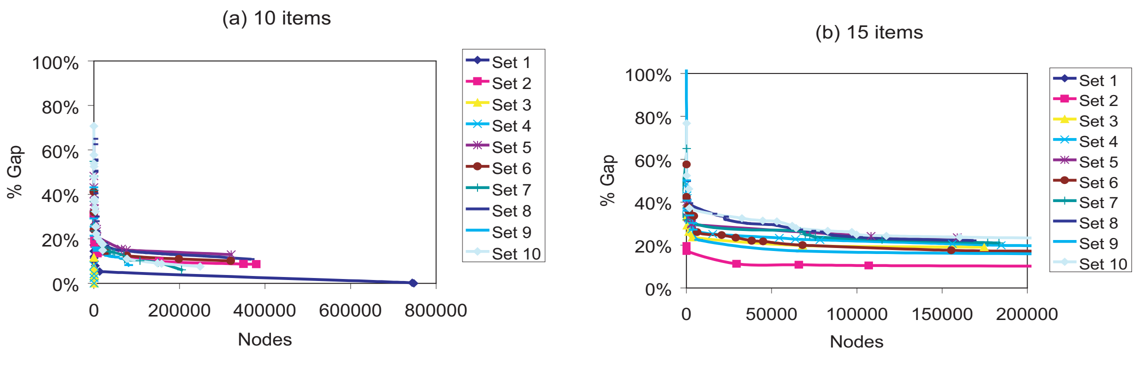

Figure 4.7: Progression of solver.

4.3 Short Summary

In the first application, we study the beneficiary-donor system of two servers and two queues. There has been a recent emphasis on studying threshold based policies for multiserver systems and many threshold policies have been proposed in the literature. We are the first to provide a general framework to compare the performance of the different threshold policies with respect to some given service metric. The optimal (state dependent) policy is obtained from the solution to a Markov decision process. The optimal threshold policy is obtained from the solution to a constrained Markov decision process using the MBEDC algorithm developed in this thesis. We show that for the weighted mean response time metric, using policies with one to three thresholds results in a close-to-optimal performance for the beneficiary-donor system of two servers and two queues. Our numerical techniques extend to the general comparison of any two threshold based policies.

In the second application, we study the multi item, two location, spare parts inventory management problem with waiting time constraints and lateral transshipments. We solve this problem exactly using the MBEDC algorithm for up to seven items easily. We can solve up to ten items but the time required is large and for three out of fifteen instances CPLEX ran out of memory because the tree file size becomes too large. For future work, it may be interesting to see if a different formulation can be developed for the particular structure of this problem, for example Formulation (F1) in chapter 2. Though this formulation was inefficient for the CDOS problem, its efficiency for the spare parts problem should be studied since this formulation does not introduce degeneracy into the mathematical program. In addition, it would be interesting to obtain the actual savings obtained from (i) lateral transshipments by comparing it to a similar model without lateral transshipments and (ii) from using a weighted target waiting time constraint over all items rather than a target waiting time constraint for each item which is stronger. The problem without lateral transshipments as well as the problem with target waiting times for each item are easier to solve for the MBEDC algorithm. The latter problem can, in fact, be separated into I single item problems, i.e., single-item constrained Markov decision process which can be solved quickly using the MBEDC algorithm.

In the next chapter, we study a revenue-management problem in the service industry, in which the seller has to negotiate the sale price with each buyer rather than posting the sale price. In addition, we study the case when the seller has to ensure that the average buyer is no worse-off from price bargaining compared to price-posting via a constraint.

Chapter 5

Revenue Management with Nash Bargaining

Born in the airline industry, revenue-management has developed into a mature decision technology to support a seller's tactical capacity-allocation and pricing decisions. Its most successful applications have been in business-to-consumer settings, such as the airline, travel, and hospitality industries, and, more recently, retailing (Bitran and Caldentey 2003, Boyd and Bilegan 2003, Elmaghraby and Keskinocak 2003). In addition, novel applications are being designed and developed in a variety of industries (Talluri and van Ryzin 2004, Chapter 10). Geoffrion and Krishnan (2001, p. 16) note that revenue-management technology, supported by e-mail and the web, "allow[s] companies to change prices [...] globally and instantly as commitment deadlines approach." Additionally, they state that it "enable[s] rapid customer responses to price changes and subsequent real-time or near-real-time negotiations". Secomandi (2006) highlights the potential usefulness of revenue-management to support negotiations in business-to-business (B2B) settings, where companies establish a variety of contracts with negotiated terms and conditions, including price, to manage the exchange of goods and services. However, the price-based revenue-management literature has only studied the posted-price case (Bitran and Mondschein 1997, Zhao and Zheng 2000, Talluri and van Ryzin 2004, Chapter 5), which does not fully capture the negotiated nature of business transactions. Therefore, there is a gap in this literature: While its models are recognized as potentially useful to support commercial negotiations, to the best of our knowledge, there is no published work that explicitly models the bargaining nature of business transactions in a revenue-management setting.

Focus. Differently from the existing revenue-management literature, the focus of this chapter is on incorporating price-bargaining features in revenue-management modeling. (We use the terms bargaining and negotiation interchangeably.) We study the classical price-based revenue-management setting of a seller that sells a finite stock of a single product within a finite number of time periods. In each period there is a positive, possibly time dependent, probability of a single buyer arrival. The inventory level and the number of time periods to go define the *state*. The seller bargains the sale price with each arriving buyer. Each buyer has a reservation value corresponding to the maximum price it is willing to pay for one unit of product. This value may depend on the alternative offers available to the buyer from the seller's competitors. For the seller, each buyer's reservation value is a random variable with known distribution. The seller has a state-dependent reservation value corresponding to the minimum price it is willing to accept for one unit of product. Since the buyers cannot observe the seller's state, they do not know the seller's reservation value.

Research questions. Our goal in this paper is to answer the following research questions (RQs).

(RQ1:) How can one model price negotiations in a stochastic and dynamic revenue-management setting?

(RQ2:) By how much does the seller's expected revenue improve from adopting a dynamic and stochastic, as opposed to a myopic and stochastic, bargaining model in this setting?

(RQ3:) Does the seller benefit from bargaining prices rather than posting them as in the traditional revenue-management case?

(RQ4:) Can both the seller and the buyers be better-off by bargaining price rather than transacting via posted-prices?

Contributions. We contribute to the science of operations in the area of revenue-management. In §5.1, we present the Nash (1950) bargaining solution, which is a static and deterministic model, in the context of a buyer-seller price negotiation. In §5.2, we embed this model of bargaining in a dynamic program that maximizes the seller's expected revenue from selling the given inventory of a product during the planning horizon. This model dynamically optimizes the seller's reservation value. We prove that in any stage the seller's optimal value function is increasing and concave in inventory. We numerically estimate the value of dynamically bargaining prices is about 6-8% higher than negotiating them in a myopic fashion. When the buyer's reservation value is normally distributed, we also show that the changes in the seller's optimal value functions and optimal reservation values are small for a small change in the variance of the buyer's reservation value, but cannot be neglected for a large change in the variance of the buyer's reservation value.

In §5.3, we compare our bargaining model against the traditional posted-price model. We prove that in any state, the seller's optimal value function is always greater in the bargaining model than in the posted-price model. We also numerically estimate the magnitude and the behavior of this improvement. We find that the improvement is substantial (14-30% on average). We analyze this increase in expected revenue by isolating the roles played by two cumulative effects: (i) quantity effect and (ii) price effect. The first effect reflects the change in the probability of selling one unit of the product and is always positive in our experiments. The second effect reflects the change in the expected net price, i.e., the transacted price minus the opportunity cost. This effect may be positive or negative but is almost always positive. We also observe that the relative improvement of the bargaining model over the posted-price model increases with higher variability of the buyer's willingness-to-pay, when this random variable is normally distributed. We also find that this improvement is rather insensitive to the probability of customer arrival.

Though the main result of §5.3 implies that the seller is better-off by bargaining rather than posting prices, the buyers can be worse-off. In §5.4, we present a modification of the dynamic bargaining model discussed in §5.2 that ensures that both the seller and the buyers are no worse-off by bargaining prices rather than transacting through price-posting. We numerically show that by using this model the seller is able to capture most of the benefit of the dynamic bargaining model presented in §5.2 even without making the buyers worse-off compared to the posted-price model.

Relevance. While we contribute to the science of operations, our models and results have potential for practical relevance: They provide foundations for prescriptive/normative modeling in revenue-management with bargaining. The significant improvement of the seller's value functions over the posted-price model provides theoretical support to the existing practices of commercial negotiations. However, businesses that adopt myopic strategies are not making optimal decisions. By adopting a dynamic strategy, expected revenue can be increased by an average of 6-8%. This points to the importance of incorporating dynamic optimization approaches into negotiation support systems, such as the one discussed by Rangaswamy and Shell (1997).

Related literature. There is a vast literature dealing with bargaining. Muthoo (1999) provides an introduction to this theory and its applications. In particular, he describes two fundamental bargaining models: the Nash (1950) bargaining solution (NBS) and the Rubinstein (1982) alternating offer model. We choose to embed the former model in a dynamic and stochastic price-based revenue-management setting because of its versatility and simplicity (see Muthoo 1999, ch. 2). We also note that Corfman and Gupta (1993) review several applications of the Nash bargaining model in the context of group choices. We are not the first to embed the Nash bargaining model within a stochastic setting where decisions are made dynamically. For example, Lippman and McCardle (2004) embed the Nash bargaining model within a decision tree to evaluate the decisions made by a heir-claimant. However, to the best of our knowledge, we are not aware of any other published paper that incorporates a bargaining model in a stochastic and dynamic price-based revenue-management model. Terwiesch, Savin and Hann (2005) do develop a formal model of haggling between a name-your-own-price retailer and a set of individual buyers, and compare the profits from their model to those of a posted-price model. However, our setting is different because buyers do not name their own prices, the seller faces a supply constraint, and uses a different reservation value to negotiate with each buyer over time. In addition, differently from these authors, we propose a model that allows the seller to capture the majority of the benefit of bargaining without making its customers worse-off with respect to the posted-price case.

5.1 Nash Bargaining Solution to Price Negotiation

Consider a buyer and a seller negotiating the sale price of a single unit of product in a single period, deterministic setting (see Figure 5.1). The buyer has a resale price r for one unit of the product and a best alternative offer from the seller's competitors, $w \leq r$. Thus, the buyer's reservation value is w, which is the maximum price it is willing to pay for one unit of the product. Similarly, the seller has some reservation value u, which is its minimum price acceptable for selling one unit of the product. If the buyer's reservation value w is less than the seller's reservation value u, then there is no sale. Otherwise, an agreement for the sale price p is reached such that $u \leq p \leq w$, and the buyer purchases one unit of product from the seller at this price.

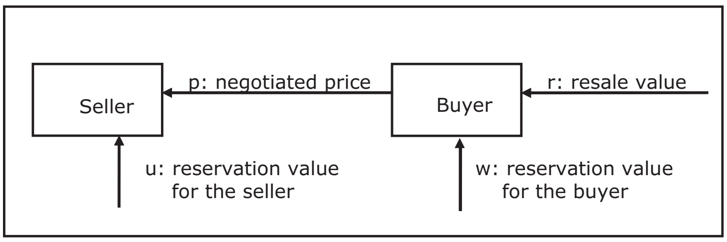

Figure 5.1: Description of the price negotiation setting.

We now show the result of applying the Nash (1950) bargaining solution formula to determine the negotiated sale price p. To do so, we assume that the seller's and its buyers' respective utilities from a sale agreement at price p are equal to the monetary values of their payoffs relative to their reservation values, i.e., $U^S(p, u) = p - u$ and $U^B(p, w) = w - p$. The values u

and w are known to both parties. The size of the pie that is to be split between the seller and the buyer is $w - u$. The minimum share of the pie that is acceptable to both the seller and the buyer is zero and, hence, their so-called disagreement values, d^S and d^B, respectively, are zero. The Nash bargaining solution $p^{NBS}(u, w)$ is then defined as follows:

$$p^{NBS}(u,w) := \arg\max_{\{p:\ p \geq 0,\ U^B(p,w) \geq d^B,\ U^S(p,u) \geq d^S\}} \left[U^B(p,w) - d^B\right]\left[U^S(p,u) - d^S\right]$$

$$= \arg\max_{\{p:\ u \leq p \leq w\}} (w-p)(p-u)$$

$$= \begin{cases} (w+u)/2 & \text{if } w \geq u \\ \text{not defined} & \text{otherwise.} \end{cases} \tag{5.1}$$

The assumption that both parties know each others' reservation values *before* entering into negotiation, i.e., ex-ante, is not true in our setting. These values become known once the two parties enter the negotiation, i.e., ex-post. From the seller's point of view, ex-ante, the buyer's reservation value is modeled as a random variable W with known distribution. We assume that $W(t) = W$ for all t, for simplicity. The seller's reservation value is state dependent. Since the buyer cannot observe the seller's state, it does not know the seller's reservation value ex-ante. We assume that the buyers do not behave strategically in choosing their reservation value. Given this model, the question to be answered is: How should the seller choose its reservation value in a stochastic and dynamic setting? We propose a model in the next section to determine the seller's optimal reservation value in any state (x, t).

5.2 Bargaining Model

We consider the dynamic problem of a seller that has to sell a finite stock X of a single product within a finite number of time periods T. We assume that the seller cannot reorder more units during this time horizon. The *state* is defined by the inventory level, x, and number of time periods to go, t ($t = T$ at the start and $t = 0$ at the end). In each period t, there is a positive probability of a single buyer arrival which may be time-dependent, $q(t)$. We assume $q(t) = q$ for simplicity but all our results hold if this assumption is relaxed. The buyer's reservation value is modeled as a random variable W with *pdf* $f(w)$ and *cdf* $F(w)$. Consider state (x, t), i.e., current inventory level is x and the number of time periods to go is t. Denote the seller's reservation value in state (x, t) by $u_t(x)$. (We drop x and t from some state dependent quantities in this paper for ease of writing, e.g., u, u^*, p, p^*, $p^N(\cdot)$. Some of these quantities are defined later when they are first introduced.) If there is no arrival in period t, which happens with probability $1-q$, the seller has no revenue in this period and the state changes to $(x,t-1)$. If there is an arrival in period t, which happens with probability q, then there are two cases: (i) If the buyer's reservation value w is lower than the seller's reservation value u, there is no sale. The seller has no revenue in that period and the state changes to $(x,t-1)$. (ii) Otherwise, the buyer and the seller agree to a sale price $p^N(u,w)$ given by the Nash bargaining solution with parameters w and u: $p^N(u,w) := (w+u)/2$. The seller has a revenue of $p^N(u,w)$ in that period and the state changes to $(x-1,t-1)$. An inherent assumption here is that the time required for the bargaining process between the seller and any buyer is less than the length of a time period. We denote the function $V_t^N(x)$ as the maximum expected revenue of the seller starting with x units of inventory at the beginning of period t. Model 1 below computes this function. Also, we define the function $\triangle V_t^N(x) := V_{t-1}^N(x) - V_{t-1}^N(x-1)$ as the seller's corresponding opportunity cost in state (x,t), $x=1,\ldots,X$, $t=1,\ldots,T$. Our bargaining model follows:

Model 1.

$$V_t^N(x) = (1-q)V_{t-1}^N(x) + q\ \max_{u\geq 0}\left[V_{t-1}^N(x)\ F(u) + \int_u^\infty \left(p^N(u,w) + V_{t-1}^N(x-1)\right) f(w)\ dw\right]$$

$$= V_{t-1}^N(x) + q\ \max_{u\geq 0}\left[\int_u^\infty \left(p^N(u,w) - \triangle V_t^N(x)\right) f(w)\ dw\right],\ \ x=1,\ldots,X,\ \ t=1,\ldots,T, \quad (5.2)$$

and boundary conditions $V_t^N(0)=0,\ \ t=1,\ldots,T$, and $V_0^N(x)=0,\ \ x=0,1,\ldots,X$.

Let $u_t^*(x)$ denote the solution to the maximization problem in (5.2), i.e., the optimal reservation value in state (x,t). Proposition 5.2.1 presents structural properties of the optimal value functions and optimal reservation values for Model 1. The proof is provided in Appendix A.1. Increasing/decreasing and concave are in the weak sense throughout.

Proposition 5.2.1. *(Properties of the bargaining model)*

(i) $V_t^N(x)$ *is increasing in* t *for any* x.

(ii) $V_t^N(x)$ *is concave in* x *for any* t.

(iii) $V_t^N(x)$ *is increasing in* x *for any* t.

(iv) If $F(\cdot)$ *is differentiable, the seller's optimal reservation value satisfies* $u_t^*(x) \geq \triangle V_t^N(x)$ *in any state* (x,t).

(v) $u_t^*(x)$ *is (a) decreasing in* x *for any* t, *and (b) increasing in* t *for any* x.

These properties are consistent with those of the traditional posted-price model, which is discussed in §5.3. We now quantify the value added by using a dynamic, rather than myopic, optimization approach.

5.2.1 Benefit from Dynamic Approach over Myopic Approach

A seller that ignores the effect of making a sale now on future sales will focus on maximizing the immediate expected revenue in the current period. We continue to assume that the seller negotiates the price with each buyer and the negotiated price is given by the Nash bargaining solution. The seller myopically optimizes its reservation value (u^M) in any state (x, t) as follows:

$$u^M = \arg\max_{u \geq 0} \int_u^\infty p^N(u, w)\ f(w)\ dw \tag{5.3}$$

From (5.3), the seller's (myopic) optimal reservation value u^M is the same in any state (x, t) and is equal to the value of u^* when $t = 1$, which is the same for all x in Model 1. In conjunction with Proposition 5.2.1-(v), this implies that $u^M \leq u^*$ in any state (x, t). Thus, firms that are myopic will systematically have a lower reservation value and under-bargain compared to firms that consider dynamic strategies. We numerically calculate the percentage increase of the optimal value functions of Model 1 over the value functions obtained by using u^M as the seller's reservation value, denoted by $V_t^M(x)$, and computed as follows:

$$V_t^M(x) = V_{t-1}^M(x) + q \int_{u^M}^\infty p^N(u^M, w)\ f(w)\ dw,\ x = 1, \ldots, X,\ t = 1, \ldots, T, \tag{5.4}$$

and boundary conditions $V_t^M(0) = 0,\ \ t = 1, \ldots, T$, and $V_0^M(x) = 0,\ \ x = 0, 1, \ldots, X$.

In the numerical examples of this chapter, we use two distributions of the buyer's reservation value - Uniform and Normal. These random variables are chosen as they provide a rich representation of the buyer's reservation value. Figure 5.2 displays the percentage improvement of $V_t^N(x)$ over $V_t^M(x)$ in the uniform distribution case with $W \sim U[0, 1]$, $q = 0.9$, $X = 10$, and $T = 10$. Similar results can be obtained when W is normally distributed. The percentage improvement of the dynamic strategy over the myopic strategy is very high when the ratio of the inventory level to the number of time periods to go is less than 0.6. This is because for these values, the number of time periods remaining is higher than the inventory remaining and the dynamic strategy realizes that it can use a higher reservation price for the seller in the current period since it can still sell the item in the future if it is not sold in the current period. The myopic strategy does not take future considerations into account and so it gives the same seller's reservation value in all states. The average percentage improvement over all states (x, t) is 6-8% for the two distributions. This is similar to values reported in the revenue-management literature for other transactional mechanisms (Talluri and van Ryzin 2004, p. 10).

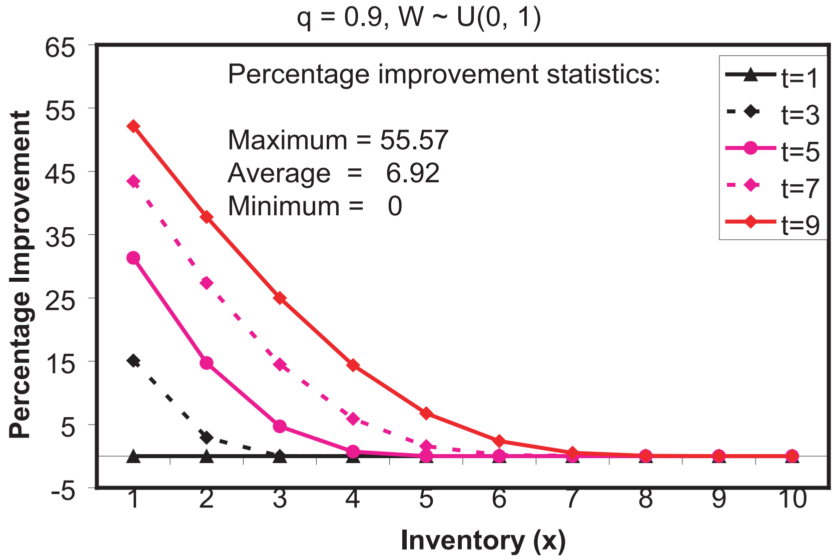

Figure 5.2: Benefit in expected revenue resulting from adopting a dynamic rather than myopic approach; W uniformly distributed.

5.2.2 Effect of Change in Variance of Buyer's Reservation Value

Next, we study the effect of changing the variance of the buyer's reservation value on the optimal value functions of Model 1 when W is normally distributed. The uniform case can be similarly analyzed. Figure 5.3a shows that the percentage change in the optimal value functions are small for a small change in the variance of the distribution of the buyer's reservation value, i.e., the standard deviation changes from 0.1 to 0.2. However, as the change in the variance becomes larger (Figures 5.3b and 5.3c), i.e., the standard deviation changes from 0.1 to 1 and from 0.1 to 2, respectively, the percentage change in the optimal value functions cannot be neglected. Similar results are obtained for any probability of customer arrival because it affects both cases of the comparisons in a similar fashion. An observation from Figure 5.3 is that when the number of time periods to go is large compared to the inventory level, it is beneficial to have higher variance in the buyer's reservation value, and lower variance otherwise. This is because the seller benefits from choosing a higher reservation value in the case when the variance is higher and the number of time periods to go is large, since if the item is not sold in the current period, it can be sold at a future time period without any loss. Lower variance is preferred in the second case because it helps to increase the probability of a sale, which is the focus in the second case.

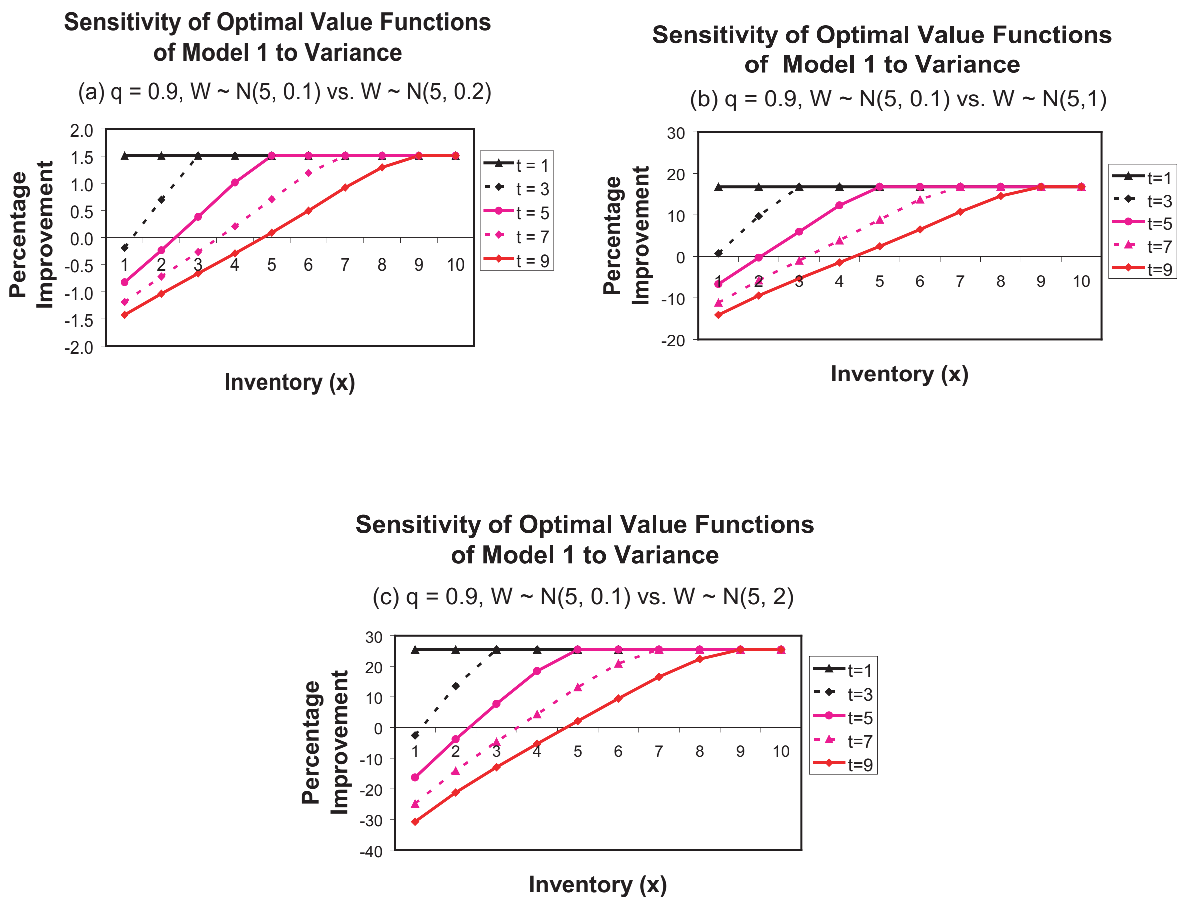

Figure 5.3: Effect of change in variance of buyer's reservation value on optimal value functions in Model 1; W normally distributed.

5.3 Bargaining Model vs. Posted-Price Model

This section compares the optimal value functions obtained from the bargaining model (Model 1) to those obtained from the traditional posted-price model (Model 2 below), first analytically for a general distribution of the buyer's reservation value, and then numerically when the buyer's reservation value is uniformly and normally distributed. Posted-price models are well studied in the revenue-management literature (Bitran and Mondschein 1997, Zhao and Zheng 2000, Talluri and van Ryzin 2004, Chapter 5). Below, we show the benchmark posted-price model for our problem setting. We assume that the buyer's reservation value does not depend on the transactional mechanism, i.e., the distribution of W is the same in Models 1 and 2.

Model 2.

$$V_t^P(x) = (1-q)V_{t-1}^P(x) + q\ \max_{p\geq 0}\left[V_{t-1}^P(x)\ F(p) + \int_p^{\infty} \big(p + V_{t-1}^P(x-1)\big)\ f(w)\ dw\right]$$

$$= V_{t-1}^P(x) + q \ \max_{p\geq 0} \left[\int_p^\infty \left(p - \triangle V_t^P(x)\right) f(w) \ dw \right], \ x = 1, \ldots, X, \ t = 1, \ldots, T, \qquad (5.5)$$

where $\triangle V_t^P(x) := V_{t-1}^P(x) - V_{t-1}^P(x-1), \ x = 1, \ldots, X, \ t = 1, \ldots, T,$
and boundary conditions $V_t^P(0) = 0, \ t = 1, \ldots, T, \ \ V_0^P(x) = 0, \ x = 0, 1, \ldots, X.$

Denote by $p_t^*(x)$ $(= p^*)$ the optimal posted-price obtained as the solution to the maximization problem in (5.5). Our main result proves that the seller's maximum expected revenue from Model 1 is always greater than that obtained from Model 2 in any state. This result is true for any distribution of the buyer's reservation value and implies that the seller should prefer to bargain rather than to adopt a posted-price model in this problem setting. It provides a theoretical justification for the current practices of bargaining in B2B settings (Geoffrion and Krishnan 2001, Secomandi 2006).

Theorem 5.3.1. *(Bargaining model vs. posted-price model)*
The seller's maximum expected revenue from Model 1 is always no less than that obtained from Model 2 in any state, i.e., $V_t^N(x) \geq V_t^P(x)$ *in any state* (x, t)*.*

Proof. (By induction.) For any $x \in \{0, \ldots, X\}$, it holds that $V_0^N(x) = V_0^P(x) = 0$, so that the property holds for $t = 0$. Assume that the property holds in period $t-1$, i.e., $V_{t-1}^N(x) \geq V_{t-1}^P(x)$ for all x. For any x, it also holds that

$V_t^N(x) = (1-q)V_{t-1}^N(x) + q \ \max_{u\geq 0} \left[V_{t-1}^N(x) \ F(u) + \int_u^\infty \left(\frac{w+u}{2} + V_{t-1}^N(x-1)\right) f(w) \ dw\right]$

$\geq (1-q)V_{t-1}^N(x) + q \ \left[V_{t-1}^N(x) \ F(p^*) + \int_{p^*}^\infty \left(\frac{w+p^*}{2} + V_{t-1}^N(x-1)\right) f(w)dw\right]$ (Since p^* is feasible)

$\geq (1-q)V_{t-1}^P(x) + q \ \left[V_{t-1}^P(x) \ F(p^*) + \int_{p^*}^\infty \left(\frac{w+p^*}{2} + V_{t-1}^P(x-1)\right) f(w) \ dw\right]$ (Induction hypothesis)

$\geq (1-q)V_{t-1}^P(x) + q \ \left[V_{t-1}^P(x) \ F(p^*) + \int_{p^*}^\infty \left(p^* + V_{t-1}^P(x-1)\right) f(w) \ dw\right]$ (Since $p^* \leq w$)

$= V_t^P(x).$

Thus, by the principle of mathematical induction, $V_t^N(x) \geq V_t^P(x)$ for all $t \in \{0, \ldots, T\}$ and $x \in \{0, \ldots, X\}$.

□

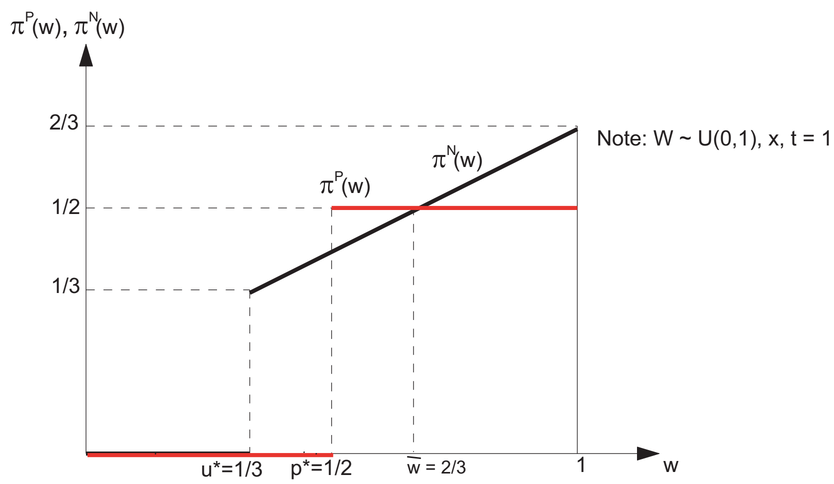

Figure 5.4: Net prices for Example 1; $W \sim U(0,1)$, $t = x = 1$, and subscript t and argument x suppressed from the relevant notation.

5.3.1 Quantity and Price Effects

We now analyze the basic result established in Theorem 5.3.1. Note that in our experiments, we find u^* is always less than p^* in all states (x, t). Hence, we discuss this case only. All the arguments extend to the case when $u^* \geq p^*$ (which never occurs in our experiments). Define the difference between the optimal bargaining and posted-price value functions as $D_t(x) := V_t^N(x) - V_t^P(x)$, for all t and x. We study this quantity by using the notion of net prices. Given a realization w of random variable W in state (x, t), define the bargaining and posted net prices, respectively, as the transacted prices minus their respective opportunity costs, if a transaction occurs, and zero otherwise:

$$\pi_t^N(x, w) \;:=\; \begin{cases} p^N(u_t^*(x), w) - \triangle V_t^N(x) & \text{if } w \geq u_t^*(x) \\ 0 & \text{otherwise} \end{cases}$$

$$\pi_t^P(x, w) \;:=\; \begin{cases} p_t^*(x, w) - \triangle V_t^P(x) & \text{if } w \geq p_t^*(x) \\ 0 & \text{otherwise.} \end{cases}$$

By Proposition 5.2.1 (iv) and the well-known result that $p_t^*(x) \geq \triangle V_t^P(x)$ (Bitran and Caldentey, 2003), the net prices are nonnegative.

Example 1 (Net prices with $W \sim U(0,1)$ and $t = x = 1$)**.** Figure 5.4 displays the net prices prices as functions of w when $W \sim U(0,1)$ and $t = x = 1$. In this case, both the bargained and posted-price opportunity costs are zero. It is easy to verify that $p^* = 1/2$ and $u^* = 1/3$, so that $\pi^N(w) = [(w+1/3)/2]1(w \geq 1/3)$ and $\pi^P(w) = (1/2)1(w \geq 1/2)$, with $1(\cdot)$ equal to 1 if its argument is true and zero otherwise, and subscript t and argument x suppressed from the notation for simplicity.

In the bargaining case, the net price is zero if $w \in [0, u_t^*(x))$, with linear payoff $p^N(u_t^*(x), w) - \triangle V_t^N(x)$ otherwise, which increases at a slope of 1/2. In contrast, in the posted price case, the net price is zero if $w \in [0, p_t^*(x))$, constant otherwise. Thus, in this case the net price for "high" willingness-to-pay realizations is equal to that corresponding to realizations that "barely" exceed $p_t^*(x) = \frac{1}{2}$. In order to compare the two net prices, define $\overline{w}$ as the point when $\pi_t^N(x,w) = \pi_t^P(x,w)$, i.e., $\overline{w} := 2[p_t^*(x) - u_t^*(x)/2 + \triangle V_t^N(x) - \triangle V_t^P(x)]$. If $\overline{w} < p_t^*$, then the bargained net price always exceeds the posted net price when a transaction occurs under bargaining, i.e., $w \in [u_t^*(x), \infty]$. On the other hand since $u_t^*(x) \leq p_t^*(x)$, if $\overline{w} \geq p_t^*$ then the bargained net price exceeds the posted net price when w is "low" or "high," i.e., $w \in [u_t^*(x), p_t^*(x)) \cup (\overline{w}, \infty]$, and it is below this net price when w takes on "intermediate values," i.e., $w \in [p_t^*(x), \overline{w}]$.

We now "measure" the net prices and their differences. Denote by $\overline{\pi}_t^N(x)$ and $\overline{\pi}_t^P(x)$, respectively, the expected values of net prices $\pi_t^N(x,w)$ and $\pi_t^N(x,w)$ with respect to the probability distribution $F(w)$ of the buyer's willingness to pay:

$$\begin{aligned}\overline{\pi}_t^N(x) &= \int_{u_t^*(x)}^{\infty} \pi_t^N(x,w) f(w) dw \\ \overline{\pi}_t^P(x) &= \int_{p_t^*(x)}^{\infty} \pi_t^P(x,w) f(w) dw.\end{aligned}$$

The seller's immediate benefit from using a bargained, rather than posted, price in state (x,t) is the difference between these two values: $\overline{\pi}_t^N(x) - \overline{\pi}_t^P(x)$. We decompose this difference into a quantity and a price effect as follows:

$$\begin{aligned}\overline{\pi}_t^N(x) - \overline{\pi}_t^P(x) &= \int_{u_t^*(x)}^{\infty} \pi_t^N(x,w) f(w) dw - \int_{p_t^*(x)}^{\infty} \pi_t^P(x,w) f(w) dw \\ &= \underbrace{\int_{u_t^*(x)}^{p_t^*(x)} \pi_t^N(x,w) f(w) dw}_{\text{Quantity effect}} + \underbrace{\int_{p_t^*(x)}^{\infty} [\pi_t^N(x,w) - \pi_t^P(x,w)] f(w) dw}_{\text{Price effect}}.\end{aligned}$$

We denote $\mathrm{QE}_t(x)$ and $\mathrm{PE}_t(x)$, respectively, the quantity and price effects in state (x,t). The interpretation of these effects is as follows. In state (x,t), given a realization w in the interval $[0, u_t^*(x))$ no sale occurs under any of the two transactional mechanisms. If $w \in [u_t^*(x), p_t^*(x))$, a

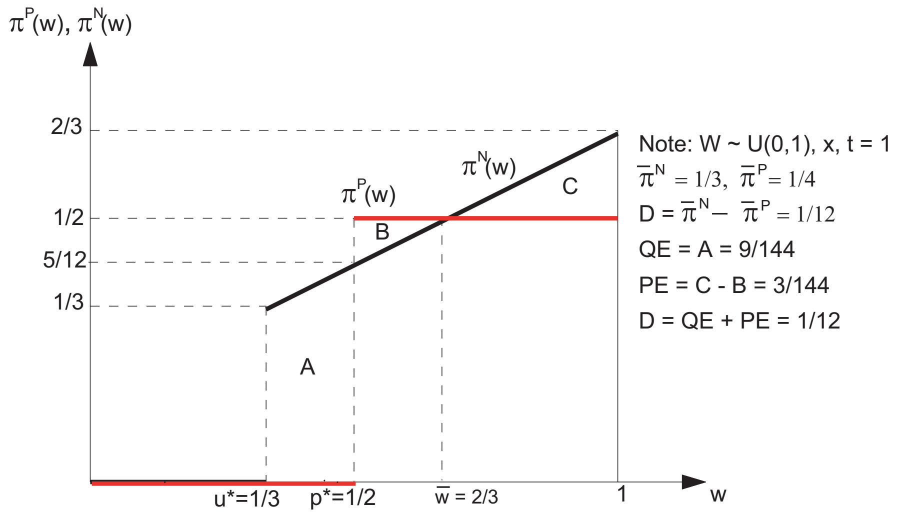

Figure 5.5: Quantity and price effects for Example 1; subscript t and argument x suppressed from the relevant notation.

sale occurs only when bargaining price, in which case the seller's net gain is $\pi_t^N(u^*(x), w)$. The quantity effect measures this payoff by weighing it by the probability density function $f(w)$. We always observe that $u_t^*(x) \leq p_t^*(x)$ and hence this effect is always observed to be nonnegative but we are unable to prove this conjecture. Instead, if $w \in [p_t^*(x), \infty]$, a sale occurs under both bargaining and posted price. In this case, the immediate benefit from bargaining is the net price difference $\pi_t^N(u_t^*(x), w) - \pi_t^P(x, w)$, whose sign is undetermined. The price effect amounts to measuring this difference according to the probability distribution $F(w)$ over the interval $[p_t^*(x), \infty]$. Hence, this effect can be positive, zero, or negative.

Example 2 (Quantity and price effects for Example 1)**.** Continuing Example 1, Figure 5.5 illustrates the quantity and price effects. It is easy to verify that $\overline{\pi}^N = 1/3$ and $\overline{\pi}^P = 1/4$, so that their difference is $D = 1/12$. This value decomposes into the sum of the quantity and price effects. The quantity effect is equal to area A, QE $= A = 9/144$, the price effect to the differences of areas C and B, PE $= C - B = 4/144 - 1/144 = 3/144$, and their sum is $12/144 = 1/12$, which is of course equal to D. This shows that in this case the quantity effect explains 75% of the immediate benefit of bargaining over posted price, the price effect the remaining 25%.

We now employ these effects to decompose quantity $D_t(x)$ in a recursive manner as follows:

$$
\begin{aligned}
D_0(x) &:= 0, \; x = 1, \ldots, X \\
D_t(x) &= V_t^N(x) - V_t^P(x)
\end{aligned}
$$

$$\begin{aligned} &= D_{t-1}(x) + q\left[\overline{\pi}_t^N(x) - \overline{\pi}_t^P(x)\right] \\ &= D_{t-1}(x) + q\left[\mathrm{QE}_t(x) + \mathrm{PE}_t(x)\right],\ t = 1, \ldots, T,\ x = 1, \ldots, X. \end{aligned}$$

Given $x = 1, \ldots, X$, it holds that $D_1(x) = q\left[\mathrm{QE}_1(x) + \mathrm{PE}_1(x)\right]$. This implies the following result.

Proposition 5.3.1. *(Value-function difference decomposition)*

$$D_t(x) \quad = \quad q \sum_{\tau=1}^{t} \left[QE_\tau(x) + PE_\tau(x)\right],\ t = 1, \ldots, T,\ x = 1, \ldots, X.$$

Thus, the value-function difference $D_t(x)$ in state (x, t), with $x = 1, \ldots, X$ and $t = 1, \ldots, T$, can be interpreted as the sum of the quantity and price effects in the remaining time periods, $\tau = t, \ldots, 1$, at the same inventory level, x, weighed by the probability that a customer arrives, q (this is one way of decomposing the value-function difference). We call these quantities the cumulative quantity and price effects in this state. This decomposition of the optimal value-function difference $D_t(x)$ allows us to study how bargaining outperforms the posted-price mechanism in all states (x, t) in terms of relative contributions of the cumulative quantity and price effects. We numerically quantify these effects in §5.3.2.

5.3.2 Numerical Examples

We numerically compute the percentage improvement of the seller's optimal value functions obtained from Model 1 over those from Model 2 for two different distributions of the buyer's reservation value - uniform and normal. Sensitivities to the variance of buyer's reservation value and to the probability of buyer arrival are studied. We also quantify the cumulative quantity and price effects for these distributions.

Comparison of value functions. The percentage improvement of the optimal value functions of Model 1 over Model 2 when $W \sim U[0, 1]$ and $W \sim N(5, 1)$ are plotted in Figures 5.6 and 5.7, respectively. In these plots, the horizontal axis corresponds to the inventory level and the vertical axis corresponds to the percentage improvement value. Two scenarios are considered: (i) High probability of buyer arrival ($q = 0.9$) and (ii) low probability of buyer arrival ($q = 0.3$). (We analyze the sensitivities of the comparisons, on average, to the probability of buyer arrival later in this subsection. Since the changes in these comparisons are small for small changes in q, considering these two scenarios is sufficient to gain insight.)

Some observations that are common to both distributions follow. (i) Since the seller can

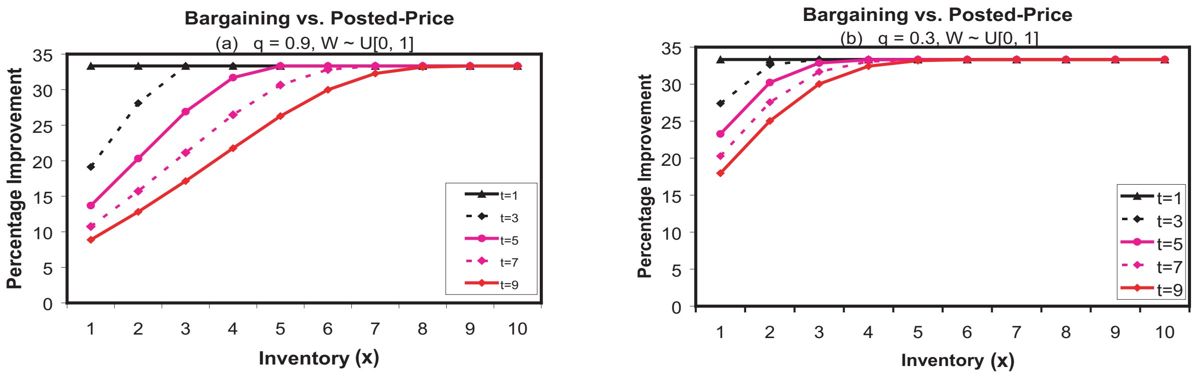

Figure 5.6: Uniform distribution - Percentage improvement of optimal value functions of Model 1 over Model 2.

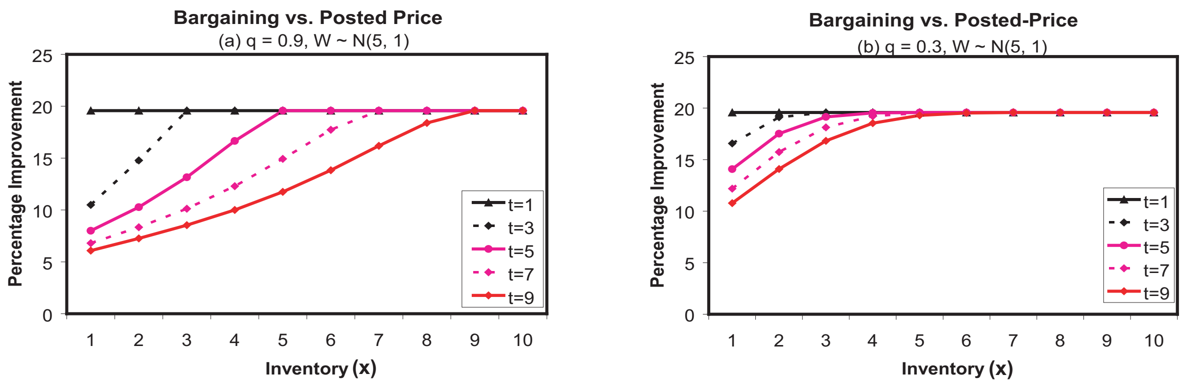

Figure 5.7: Normal distribution - Percentage improvement of optimal value functions of Model 1 over Model 2.

sell at most one unit in one period, the percentage improvement is constant when the ratio of inventory level to the number of remaining periods is one or above. (ii) The percentage improvement decreases in the number of remaining periods for a constant inventory level because both u^* and p^* approach w^{max} as this happens. (iii) The percentage improvement increases in inventory level for a constant number of remaining periods until the ratio of inventory to number of periods remaining reaches one. (iv) The percentage improvement increases as the probability of customer arrival decreases. One possible explanation is that the seller realizes a higher probability of making a sale in the bargaining model compared to the posted-price model, and as the probability of customer arrival decreases, the cumulative quantity effect becomes the major contributor to the value function difference. These observations imply that the seller's preference for bargaining over posted-price is greater if the inventory is high or with few periods remaining since the percentage improvement is the highest in such situations.

Sensitivity to buyer's reservation value variability. For the normal distribution case, we study the sensitivity of these percentage improvements with respect to the standard deviation of the buyer's reservation value while keeping the mean constant at 5. We consider three different values, 2, 1, and 0.1 for the standard deviation of the buyer's reservation value, corresponding to coefficients of variation (CV) equal to 0.4, 0.2, and 0.02, respectively. Figure 5.8 ($q = 0.9$) shows that as the variability of the buyer's reservation value increases, the percentage improvement of the seller's optimal value functions of Model 1 over those of Model 2 increases. It can be numerically verified that this continues to hold for any probability of customer arrival. Thus, the more uncertain the seller is regarding the buyer's reservation value, the more it should prefer to bargain than to adopt a posted-price policy. The reason is that in any period, the seller can obtain a higher price from buyers with high reservation values in bargaining as compared to posting-prices without decreasing the probability of sale agreement.

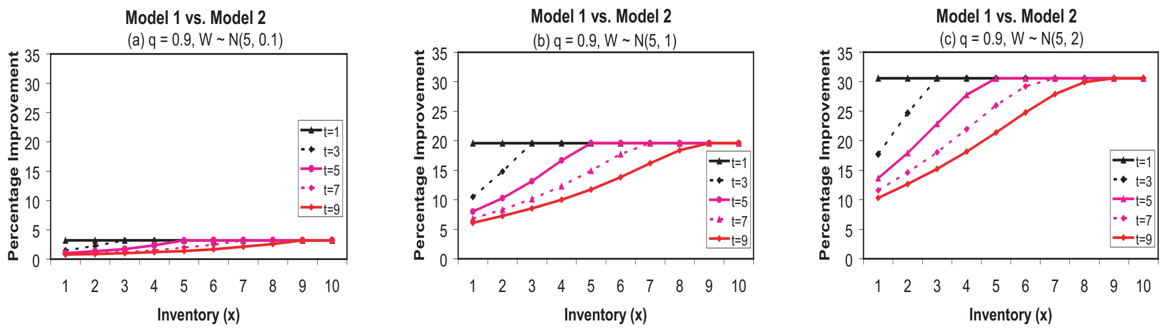

Figure 5.8: Effect of variance of buyer's reservation value on percentage improvement of optimal value functions of Model 1 over Model 2: $q = 0.9$, (a) CV = 0.02 (b) CV = 0.2 (c) CV = 0.4. Thus, variance of the buyer's reservation value distribution increases from left to right.

Sensitivity to probability of customer arrival. We now study the effect of probability of buyer arrival (q) on the average percentage improvement of the optimal value functions of Model 1 over those of Model 2. The average percentage improvement values taken over all states (x, t) as a function of q are shown in Figure 5.9. For both the uniform and normal distributions, we find that the average percentage improvement values decrease as q increases, but the decrease is small for small changes in q because the probability of customer arrival, q, has similar effects on both cases.

Quantification of cumulative quantity and price effects. Figures 5.10 and 5.11 show the cumulative quantity and price effects for the uniform and normal distributions, respectively, with $q = 0.9$, and $X = T = 10$. Similarly, Tables 5.1 and 5.2 give the percentage contribution of the cumulative quantity effect to the difference between the optimal value functions of the two models, for the two distributions, respectively. The percentage contribution of the cumulative price effect can be easily obtained from these values.

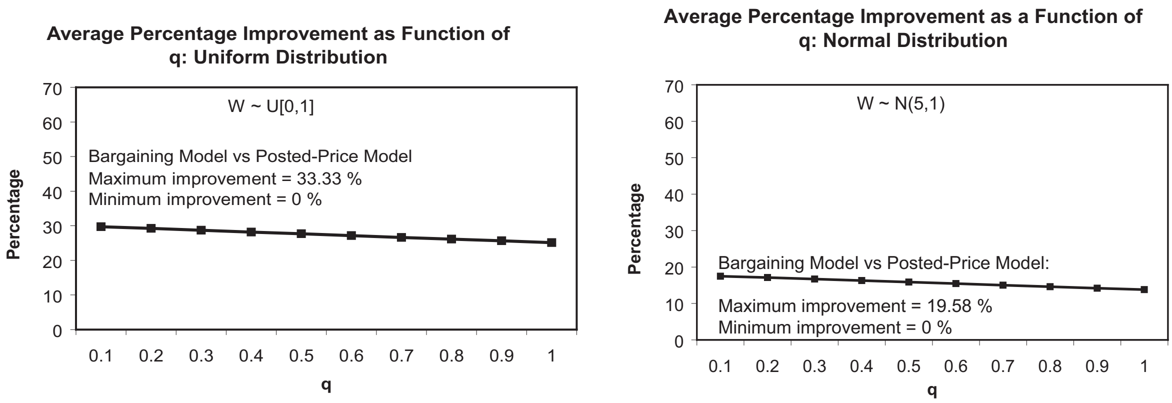

Figure 5.9: Average percentage improvement of optimal value functions of Model 1 over Model 2 as a function of q.

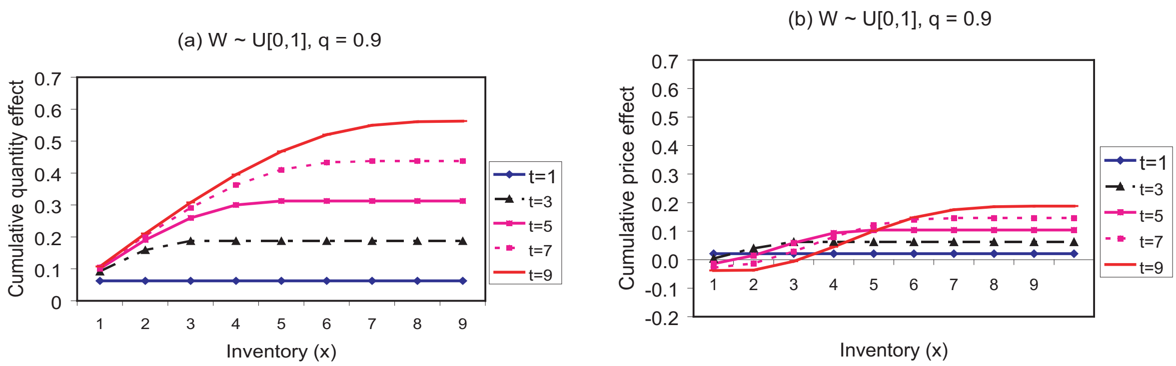

Figure 5.10: Cumulative quantity and price effects, $W \sim U[0,1]$, $q = 0.9$, $X = T = 10$.

Inven-tory(x)	Time to go (t)									
	1	2	3	4	5	6	7	8	9	10
1	**75**	**86**	**96**	**106**	**116**	**126**	**136**	**146**	**156**	**165**
2	**75**	**75**	**79**	**86**	**93**	**100**	**107**	**114**	**121**	**128**
3	**75**	**75**	**75**	**78**	**81**	**86**	**91**	**96**	**102**	**108**
4	**75**	**75**	**75**	**75**	**76**	**79**	**82**	**86**	**90**	**94**
5	**75**	**75**	**75**	**75**	**75**	**76**	**77**	**80**	**82**	**85**
6	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**76**	**78**	**80**
7	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**76**	**77**
8	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**76**
9	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**
10	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**	**75**

Table 5.1: Percentage contribution of the cumulative quantity effect to total improvement, $W \sim U[0,1]$, $q = 0.9$, $X = T = 10$.

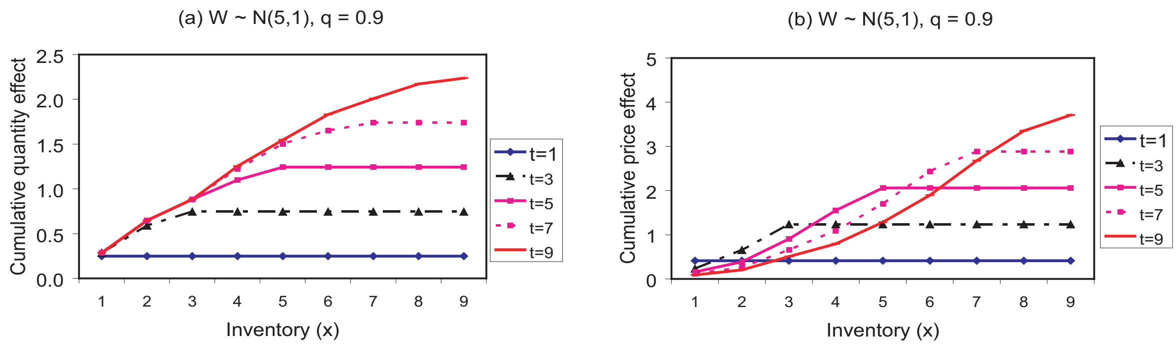

Figure 5.11: Cumulative quantity and price effects, $W \sim N(5,1)$, $q = 0.9$, $X = T = 10$.

Inven-tory(x)	Time to go (t) 1	2	3	4	5	6	7	8	9	10
1	38	49	**55**	**60**	**65**	**68**	**71**	**74**	**76**	**78**
2	38	38	47	**55**	**63**	**67**	**71**	**74**	**76**	**78**
3	38	38	38	44	49	**53**	**57**	**61**	**64**	**66**
4	38	38	38	38	41	48	**53**	**58**	**61**	**64**
5	38	38	38	38	38	42	47	**51**	**54**	**58**
6	38	38	38	38	38	38	40	44	49	**53**
7	38	38	38	38	38	38	38	40	42	47
8	38	38	38	38	38	38	38	38	39	42
9	38	38	38	38	38	38	38	38	38	40
10	38	38	38	38	38	38	38	38	38	38

Table 5.2: Percentage contribution of the cumulative quantity effect to total improvement, $W \sim N(5,1)$, $q = 0.9$, $X = T = 10$.

We observe that for the uniform distribution case, the cumulative quantity effect is the major contributor to the percentage improvement of the optimal value functions in all states. In fact, the cumulative price effect may even be negative (entries above 100% in Table 5.1 in some states). The results are different for the normal distribution. The major contributor is the cumulative price effect when the ratio of inventory to number of remaining periods is greater than or equal to one, i.e., the unconstrained inventory case. When this ratio is between zero and one, i.e., the constrained inventory case, the quantity effect is typically the major contributor. We explain these observations by studying the unconstrained ($x \geq t$) and constrained ($x < t$) inventory cases.

Case 1 (Unconstrained inventory with $x \geq t$)**.** In this case, the opportunity costs in both the bargaining and posted-price models are zero , i.e., $\triangle V_t^N(x) = \triangle V_t^P(x) = 0$. For the uniform distribution example ($W \sim U[0,1]$), it is easy to show that $u_t^*(x) = \frac{1}{3}$, $p_t^*(x) = \frac{1}{2}$, $V_t^N(x) = \frac{qt}{3}$, $V_t^P(x) = \frac{qt}{4}$, and $D_t(x) = V_t^N(x) - V_t^P(x) = \frac{qt}{12}$. Also, $QE_t(x) = \frac{1}{16}$ and $PE_t(x) = \frac{1}{48}$. Thus, the cumulative quantity effect ($\sum_{\tau=1}^t QE_t(x)$) is thrice the cumulative price effect ($\sum_{\tau=1}^t PE_t(x)$) and hence contributes 75% of the increase in the optimal value functions of the bargaining model over the posted-price model. For the normal distribution example ($W \sim N(5,1)$), we numerically obtain $u_t^*(x) = 3.50$, $p_t^*(x) = 3.90$, $V_t^N(x) = 3.63t$, $V_t^P(x) = 3.03t$, and $D_t(x) = V_t^N(x) - V_t^P(x) = 0.60t$. Also, $QE_t(x) = 0.25$ and $PE_t(x) = 0.41$. Thus, the cumulative price effect is almost twice the cumulative quantity effect and contributes 62.38% of the increase in the optimal value functions of the bargaining model over the posted-price model. Note that the cumulative price effect is the major contributor to the value function difference for the normal distribution examples, while the cumulative quantity effect is the major contributor in the uniform distribution example. One explanation is that the probability mass in the tail of the distribution relative to p^* is greater in the normal distribution example (=0.864) compared to the uniform distribution example, (=0.5) and hence the cumulative price effect has a more dominant role in the former case.

Case 2 (Constrained inventory with $x < t$)**.** The opportunity costs in both models are non-zero in such states, and hence it is difficult to obtain a closed form formula for the cumulative effects and the value functions difference that explains the relative strengths of the two effects, even for the case when $x = 1$ and $t = 2$. However, we observe that as the opportunity cost increases (as t increases or as x decreases), the relative contribution of the cumulative quantity effect to the difference in the value functions of the two models increases for both the uniform and normal distributions. This says that when the inventory is constrained the portion of the value function difference that can be attributed to the revenue generated by selling under bargaining and not selling under posted price (since we know $u^* \leq p^*$) outweighs that attributable to the revenue generated by selling in both cases, albeit at two different prices. This is counterintuitive because with constrained inventory one would expect price to be a more important factor than quantity. One explanation is that as opportunity cost increases, both u^* and p^* increase such that the probability mass in the tail of the distribution (relative to p^*) decreases more rapidly than the probability mass between u^* and p^*.

5.4 Pareto Improvement on Posted-Price Model

As shown in §5.3, Model 1 outperforms Model 2 with respect to the seller's optimal value function. While the seller may have multiple buyers, we represent them by an aggregate buyer and compute an aggregate buyers' value function, $B_t(x)$ (as defined in (5.7) and (5.9) later in this section). It can be shown numerically that in any state (x,t), $B_t(x)$ is smaller when the seller uses Model 1, rather than Model 2, for both distributions of the buyer's reservation value discussed in §5.3. This may raise some concerns regarding the fairness of Model 1 to both parties. The question is whether a model exists that yields a Pareto improvement over the posted-price model, for any distribution of the buyer's reservation value. Figure 5.12 describes this situation. Select any state (x,t). The buyers' and the seller's value functions in this state for the posted-price model are given by $B_t^P(x)$ and $V_t^P(x)$, respectively. We take this to be the origin of a two dimensional-space whose horizontal and vertical axes, respectively, represent the buyers' and seller's value functions in this state for any model. Any model that results in moving to the bottom-left/right quadrants is less preferable than the posted-price model. Hence, these regions are uninteresting and are crossed out. From our experiments, we know that Model 1 typically results in a movement to the top-left quadrant, i.e., the seller does better compared to the posted-price model but the buyer is worse-off. However, we cannot exclude improvements to the top-right quadrant for this model. Below, we consider a hybrid bargaining - posted-price model (Model 3) that results in an at least weak movement to the top-right quadrant for any distribution of the buyer's reservation value when such movements exist, else it remains at the origin. The main idea is that the seller chooses its disagreement value such that the desired improvements occur. In numerical experiments such a movement always exists.

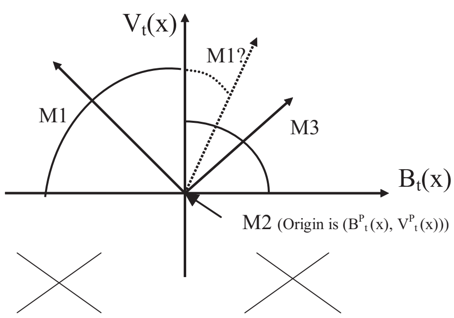

Figure 5.12: Pareto improvement with respect to posted-price model in state (x,t).

Suppose the optimal value function values for the posted-price model as well as the corresponding optimal posted-prices are known for all states, i.e., $V_t^P(x)$, $B_t^P(x)$, $p_t^*(x)$ are known for all states. For this model, define the seller's and buyer's continuation value functions, $S_t^P(x)$ and $R_t^P(x)$, respectively, for $x = 1, \ldots,$ X and $t = 1, \ldots, T$, as follows:

$$S_t^P(x) := V_{t-1}^P(x)\ F(p^*) + \int_{p^*}^{\infty} \left(p^* + V_{t-1}^P(x-1)\right) f(w)\ dw, \tag{5.6}$$

$$R_t^P(x) := B_{t-1}^P(x)\ F(p^*) + \int_{p^*}^{\infty} \big(w - p^* + B_{t-1}^P(x-1)\big)\, f(w)\ dw, \tag{5.7}$$

with boundary conditions $V_t^P(0) := B_t^P(0) := 0$, $t = 1, \ldots, T$, and $V_0^P(x) := B_0^P(x) := 0$, $x = 0, 1, \ldots, X$.

Incidentally note that with this notation, for $x = 1, \ldots, X$, and $t = 1, \ldots, T$, it follows that

$$V_t^P(x) = (1-q)V_{t-1}^P(x) + qS_t^P(x), \tag{5.8}$$

$$B_t^P(x) = (1-q)B_{t-1}^P(x) + qR_t^P(x). \tag{5.9}$$

Denote $V_t^H(x)$ and $B_t^H(x)$ the seller's optimal value function and the corresponding buyers' value function, respectively, in state (x,t), when the seller negotiates the price with the buyers by choosing its disagreement value $u_t(x)$ in state (x,t) based on a Pareto-improvement model. This model as follows:

Model 3.

$$V_t^H(x) = (1-q)V_{t-1}^H(x) + q\ \max\nolimits_{u \in \mathcal{C}_t(x)}\ S_t^H(x,u), \quad x = 1, \ldots, X, \quad t = 1, \ldots, T, \tag{5.10}$$

$$B_t^H(x) = (1-q)B_{t-1}^H(x) + q\ R_t^H(x,\hat{u}), \quad x = 1, \ldots, X, \quad t = 1, \ldots, T. \tag{5.11}$$

$$S_t^H(x,u) = V_{t-1}^H(x)\ F(u) + \int_u^{\infty} \left(\frac{w+u}{2} + V_{t-1}^H(x-1)\right) f(w)\ dw, \quad \begin{array}{c} x = 1, \ldots, X, \\ t = 1, \ldots, T, \end{array} \tag{5.12}$$

$$R_t^H(x,u) = B_{t-1}^H(x)\ F(u) + \int_u^{\infty} \left(w - \frac{w+u}{2} + B_{t-1}^H(x-1)\right) f(w)\ dw, \quad \begin{array}{c} x = 1, \ldots, X, \\ t = 1, \ldots, T, \end{array} \tag{5.13}$$

$$\text{where, } \mathcal{C}_t(x) = \left\{u \geq 0 : S_t^H(x,u) \geq S_t^P(x), R_t^H(x,u) \geq R_t^P(x)\right\}, \quad x = 1, \ldots, X, \quad t = 1, \ldots, T, \tag{5.14}$$

$$\hat{u}_t(x) := \text{argmax}_{u \in \mathcal{C}_t(x)} \; S_t^H(x,u), \quad x = 1, \ldots, X, \quad t = 1, \ldots, T, \tag{5.15}$$

and boundary conditions $V_t^H(0) = B_t^H(0) = 0$, $t = 1, \ldots, T$, and $V_0^H(x) = B_0^H(x) = 0$, $x = 0, 1, \ldots, X$.

From (5.14), $\mathcal{C}_t(x)$ is the set of feasible values of u such that the resulting buyer and seller continuation value functions are at least what they would have been if the seller adopted a posted-price strategy. From (5.10)-(5.11), it is obvious that both the seller and buyer are no worse-off compared to the posted-price model. This is true as long as $\mathcal{C}_t(x)$ is non-empty for all states, which we conjecture to be true but are unable to prove. However, this conjecture is always true in our numerical experiments.

To handle the case in which $\mathcal{C}_t(x)$ could be empty, we let the seller adopt a posted-price strategy in those states for which $\mathcal{C}_t(x)$ is empty, i.e., the seller has a hybrid strategy of either bargaining or using posted-price in every state. This can be seen as a combination of Models 1 and 2. The seller always negotiates by choosing its optimal disagreement value, $\hat{u}_t(x)$, and there is always a (weak) Pareto-improvement over the posted-price model. Figure 5.13 provides results for the case when the buyer's reservation value is uniformly distributed. Results for the normal distribution are similar.

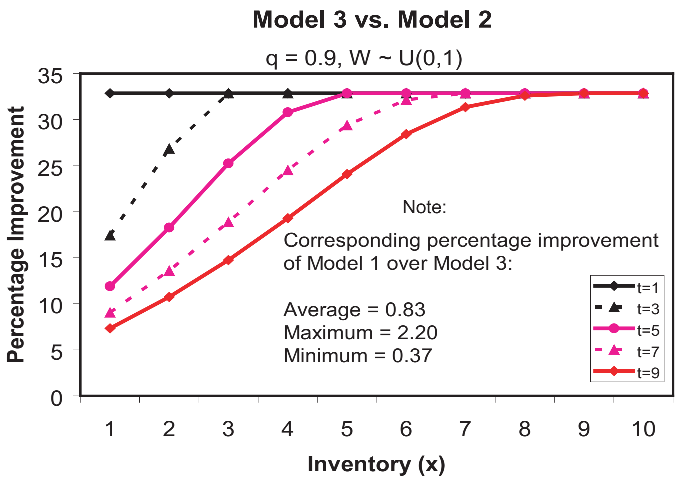

Figure 5.13: Uniform distribution - Percentage improvement of optimal value functions of Model 3 over Model 2.

Since the seller maximizes its expected revenue subject to a constraint on the buyers' expected value not being worse off with respect to Model 2, the solution results in the same values of the buyers' value functions as in the posted-price model. Thus, Model 3 provides a vertical movement from the origin in Figure 5.12. As seen from Figure 5.13, the seller's value functions are significantly higher in the hybrid model compared to the posted-price model, but negligibly lower than those of Model 1 (see note in this Figure). This is because in all states (x, t) the buyers' value in Model 1 is not too far away from the buyers' value in Model 2. The reason is that while the buyers with "high" reservation values have lower utility in Model 1 compared to Model 2, the reverse is true for buyers with reservations values between $u_t^*(x)$ and $2p_t^*(x) - u_t^*(x)$. Thus, the loss of expected revenue to the seller in adopting Model 3 as opposed to Model 1 is small compared to the possible advantages of keeping its buyers as satisfied with respect to their value functions as in the posted-price model.

5.5 Short Summary

To the best of our knowledge, our bargaining model is the first to consider price negotiation in a revenue-management setting, i.e., a finite amount of inventory must be sold in a finite number of time periods given that the seller faces uncertainty regarding the probability of a buyer arrival in a period as well as the reservation value of each buyer. We propose a model in which the seller dynamically optimizes its reservation value, establish structural properties of this model, and prove that the seller has higher expected optimal value functions in any state compared to the case when it adopts a price-posting policy. We also provide an explanation for this increase by studying the cumulative price and quantity effects. Finally, we study a model which results in a Pareto improvement in terms of the seller's optimal value functions and the average buyer value functions over the posted-price setting.

In the next chapter, we present additional numerical experiments for our bargaining model presented in this chapter. We also discuss the concept of ex-post efficiency of a bargaining model and show that our bargaining model is not ex-post efficient. We numerically show that if the seller adopts an ex-post efficient bargaining model, the decrease in its expected revenue can be substantial compared to our bargaining model.

Chapter 6

Additional Experiments for Chapter 5

In this chapter, we present additional numerical experiments conducted for the bargaining model presented in §5.2. In addition to the uniform and normal distributions for the buyer's reservation value W, we consider the exponential distribution ($W \sim exp(\lambda)$), and the triangular distribution ($W \sim Triang(0, 0.5, 1)$). These four distributions are chosen as they provide a rich representation of the buyer's reservation value for the continuous case. §6.1 shows the benefit from adopting a dynamic as opposed to a myopic approach for the triangular distribution. It also presents the percentage improvement of the optimal value functions of Model 1 over those of Model 2 for the exponential and triangular distributions. These results are similar to those obtained in §5.2.1 and §5.3.2, respectively. We study in §6.2, two relevant factors for the expected immediate reward in any state: (i) The change in the probability of reaching a sale agreement in any state, and (ii) the percentage improvement of the expected Nash price conditional on sale agreement with respect to the posted-price in any state. We observe that both these factors are positive for the bargaining model over the posted-price model. Thus, the expected immediate reward in any state is also greater in the former model. Finally, an important concept in the bargaining literature is called ex-post efficiency. A bargaining outcome is ex-post efficient if and only if after all the information is revealed, the players' payoff associated with the bargaining outcome are Pareto-efficient. This is different from the concept of Pareto improvement considered §5.4 since we are not necessarily trying to improve both the seller's and the average buyer's value function but rather considering every single negotiation and making sure each and every one of them is Pareto efficient. In §6.3, we show that the bargaining models considered in chapter 5 are not ex-post efficient, and that the only ex-post efficient model under the Nash bargaining solution assumption is the one in which the seller uses its opportunity cost as its reservation value. We show that this approach can result in significant decrease in the optimal value functions compared to the bargaining model of chapter 5. However, even in this case the optimal value functions are greater than those of the posted-price model for some distributions of the buyer's reservation value, and smaller for other distributions.

6.1 Numerical Examples for Exponential and Triangular Distributions

Benefit from dynamic approach over myopic approach.

We numerically calculate the percentage increase of the optimal value functions of Model 1 over the value functions computed using (5.3)-(5.4), assuming $W \sim Triang(0, 0.5, 1)$. Figure 6.1 displays this percentage improvement with $q = 0.9$, $X = 10$, and $T = 10$.

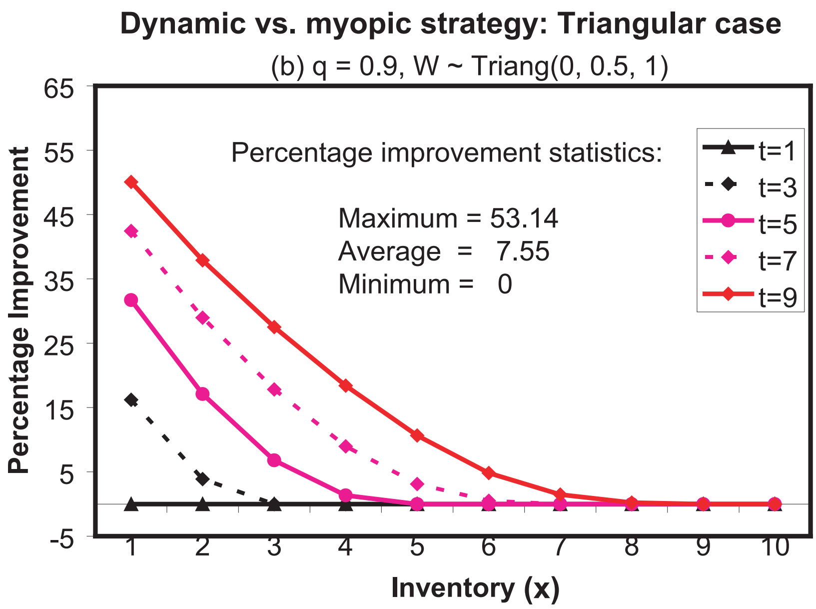

Figure 6.1: Benefit in expected revenue resulting from adopting a dynamic rather than myopic approach; W triangularly distributed.

Once again, we observe that the percentage improvement of the dynamic strategy over the myopic strategy is very high when the ratio of the inventory level to the number of time periods to go is less than 0.6. The average percentage improvement over all states (x, t) is 7.55%.

Bargaining model vs. posted-price model.

We numerically compute the percentage improvement of the seller's optimal value functions obtained from Model 1 over those from Model 2 for two different distributions of the buyer's reservation value - exponential and triangular. These values are plotted in Figures 6.2 and 6.3, respectively. In these plots, the horizontal axis corresponds to the inventory level and the vertical axis corresponds to the percentage improvement value. Two scenarios are considered: (i) High probability of buyer arrival ($q = 0.9$) and (ii) low probability of buyer arrival ($q = 0.3$). The observations are similar to those in §5.3.2 and are not repeated here. One observation unique to the exponential distribution is that the results are independent of the parameter λ. This result can be easily obtained analytically.

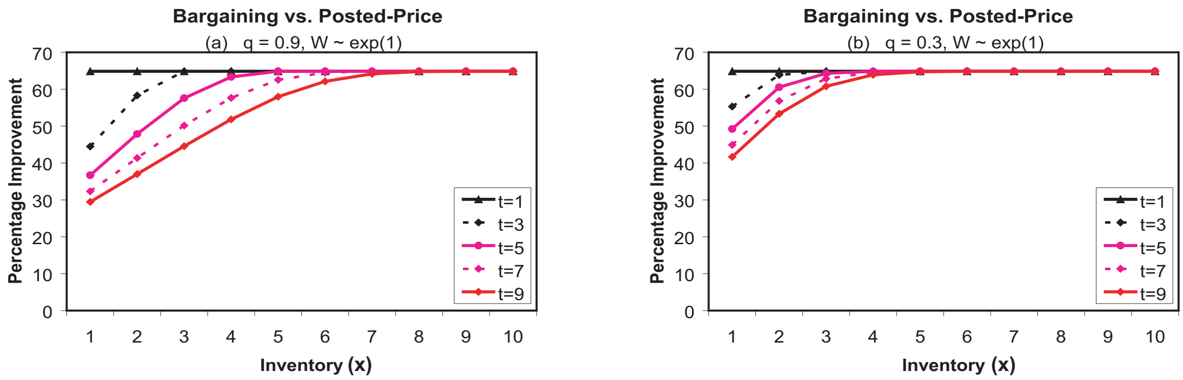

Figure 6.2: Exponential distribution - Percentage improvement of optimal value functions of Model 1 over Model 2.

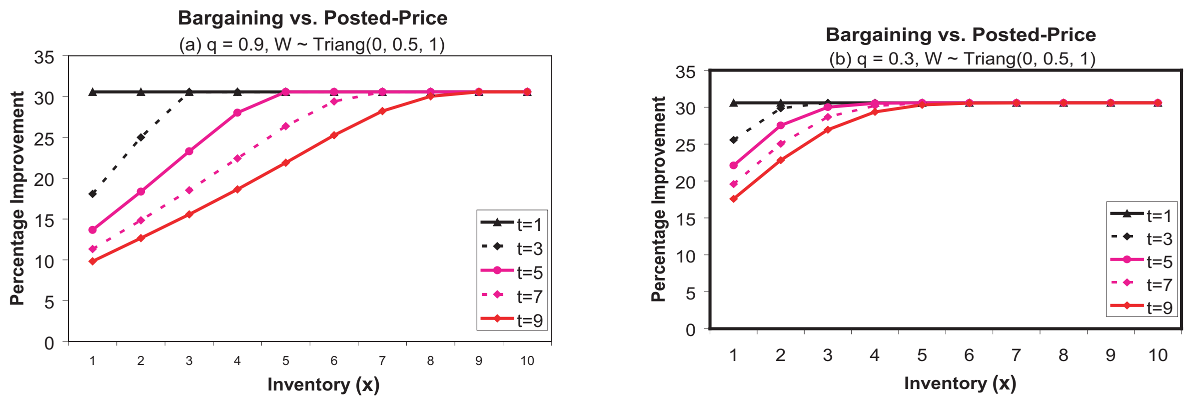

Figure 6.3: Triangular distribution - Percentage improvement of optimal value functions of Model 1 over Model 2.

Sensitivity to probability of customer arrival.

We now study the effect of probability of buyer arrival (q) on the average percentage improvement of the optimal value functions of Model 1 over those of Model 2 for the exponential and triangular distributions. The average percentage improvement values taken over all states (x, t) as a function of q are shown in Figure 6.4. For both the distributions, we find that the average percentage improvement values decrease as q increases, but the decrease is small for small changes in q.

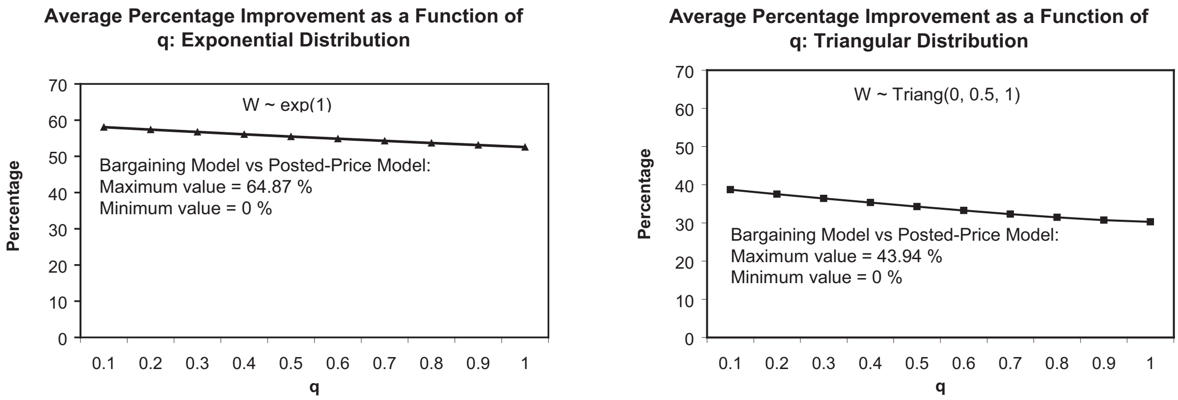

Figure 6.4: Average percentage improvement of optimal value functions of Model 1 over Model 2 as a function of q.

6.2 Factors Effecting Expected Immediate Revenue

Two reasons why the seller's expected immediate revenues in Model 1 are greater than those of Model 2 are: (i) The probability of sale agreement, given by $\bar{F}(a) = \int_a^\infty f(x)\,dx$, is higher in the former model, and (ii) the expected Nash price conditional on there being a sale in Model 1 is higher than the optimal posted-price obtained from Model 2, where the conditional expected Nash price in state (x, t) is $p_t^{NC}(x) = \left[\int_{u^*}^\infty \left(\frac{w+u^*}{2}\right)\ f(w)\ dw\right] / \bar{F}(u^*)$.

Conjecture 6.2.1.

(a) The probability of sale agreement in any state (x, t) is no smaller under Model 1 compared to Model 2. This is equivalent to $u_t^*(x) \leq p_t^*(x)$.

(b) The expected Nash price conditional on sale agreement, $p_t^{NC}(x)$, is higher than the optimal posted-price, $p_t^*(x)$, in any state (x, t).

Even though we cannot prove this conjecture, we test it numerically. Below, we evaluate both of these factors for the (previously mentioned) four distributions of the buyer's reservation value.

Probability of agreement. Table 6.1 provides results for the average *absolute* increase in the value of the probability of sale agreement in Model 1 over Model 2 for these distributions. The average increase is significant for all number of time periods.

R.V.	**Stat.**	$t = 1$	2	3	4	5	6	7	8	9	10
	min	0.17	0.08	0.05	0.03	0.02	0.02	0.02	0.01	0.01	0.01
$U[0, 1]$	avg	0.17	0.16	0.15	0.14	0.13	0.12	0.11	0.11	0.10	0.09
	max	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
	min	0.24	0.09	0.05	0.03	0.02	0.02	0.01	0.01	0.01	0.01
$exp(1)$	avg	0.24	0.22	0.21	0.20	0.19	0.17	0.16	0.15	0.14	0.13
	max	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
	min	0.07	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$N(5, 1)$	avg	0.07	0.07	0.06	0.06	0.05	0.05	0.04	0.04	0.03	0.03
	max	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
	min	0.13	0.06	0.04	0.02	0.01	0.02	0.01	0.00	0.01	0.00
$T(0, 0.5, 1)$	avg	0.13	0.12	0.12	0.11	0.10	0.09	0.08	0.07	0.07	0.06
	max	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13

Table 6.1: Absolute increase in the probability of sale agreement in Model 1 over Model 2 with $q = 0.9$.

Expected Nash price conditional on selling. Figure 6.5 provides results for the percentage improvement of the expected Nash price conditional on selling with respect to optimal posted-price in different states. There are many interesting observations from these experiments:

- The expected Nash price conditional on sale agreement (p^{NC}) is always greater than or equal to the optimal posted-price (p^*). Also, the average improvement taken over all states (x, t) is significant for all distributions.
- The percentage improvement of p^{NC} over p^* is higher when u^* is close to p^* (almost no difference in the probability of agreement) compared to when $u^* << p^*$ (much higher probability of agreement in Model 1). Thus, the two factors complement each other.
- For the uniform and exponential distributions, the percentage improvement is the highest when the inventory level is in the intermediate range (3 to 5), and lowest when the inventory level is high (7 to 10) for all times t. When the ratio of inventory level to time periods to go is greater than 0.8, the percentage improvement is small (or $u^* << p^*$). Also, with only one time period to go the expected Nash price conditional on selling and the optimal posted-price are the same and hence, the percentage improvement in this case is 0. The results are independent of λ for the exponential distribution.
- For the normal and triangular distributions, the percentage improvement with one time period to go is non-zero. This is different from the uniform and exponential distributions in which this value is zero. In fact for these cases, the improvement is significant for all states (x, t) and is lowest when the ratio of inventory level to time periods to go is less than 0.3. The reason for this difference between the distributions is not clear.

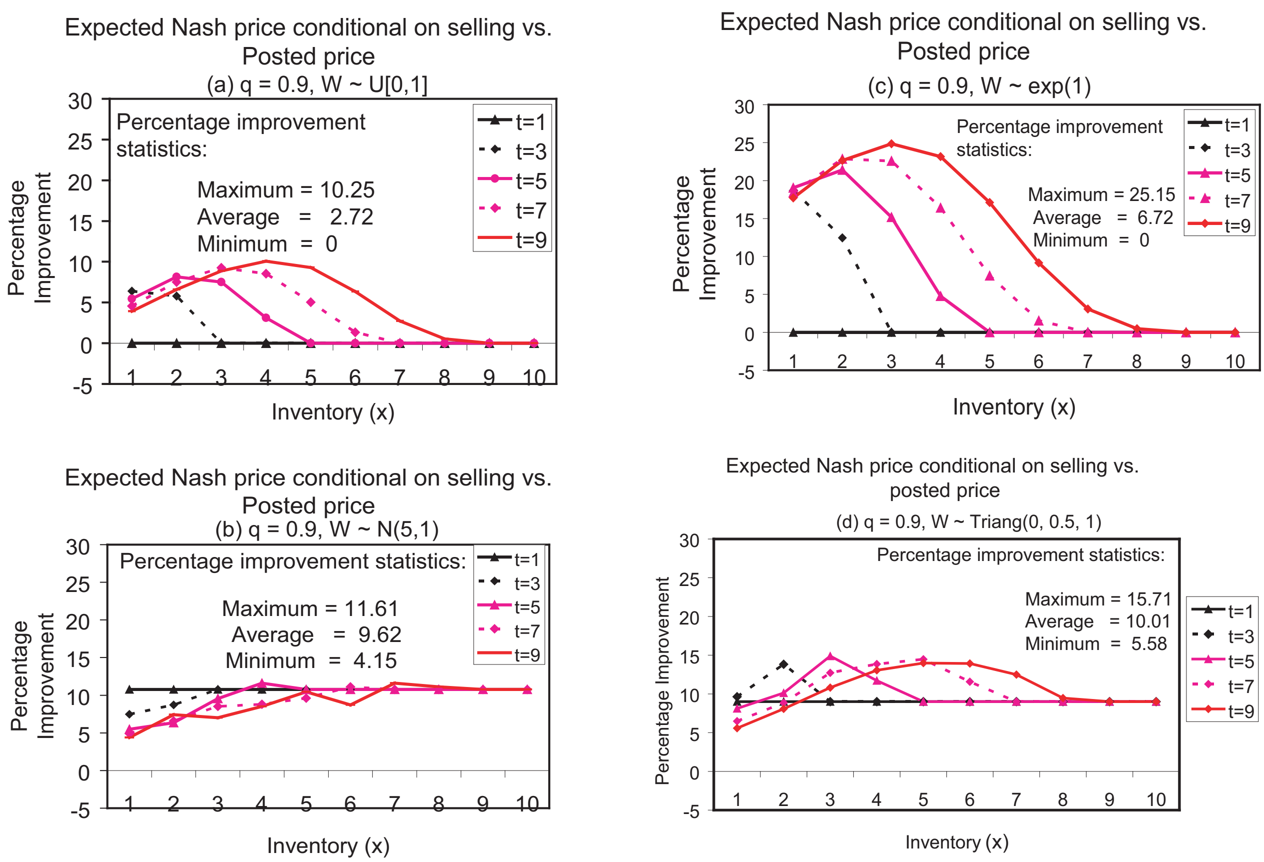

Figure 6.5: Percentage improvement of the expected Nash price conditional on selling from Model 1 compared to the optimal posted-price from Model 2.

6.3 Cost of Ex-post Efficiency

According to Muthoo (1999), a bargaining outcome is ex-post efficient if and only if after all of the information is revealed, the players' payoff associated with the bargaining outcome are Pareto-efficient. The concept of ex-post efficiency is also known as full-information efficiency. Please refer to chapter 5 for Model 1 (optimization-based bargaining model). In Model 1, if the buyer has a reservation value between the seller's opportunity cost and the seller's optimal reservation value then there is no sale which is not a Pareto-efficient result. Thus, this model is not ex-post efficient. An ex-post efficient model should result in a sale whenever the buyer's reservation value is higher than the seller's opportunity cost and no sale otherwise. Given that the bargaining outcome is modeled using a Nash bargaining solution, the only ex-post efficient model is one in which the seller uses its opportunity cost as its reservation value. We present this ex-post efficient bargaining model that uses the seller's opportunity cost as its reservation value as a second benchmark. Define $V_t^O(x)$ as the seller's optimal value function in state (x, t) for this model.

Model 4.

$$V_t^O(x) = (1-q)V_{t-1}^O(x) + q\left[V_{t-1}^O(x)\ F(\triangle V_t^O(x)) + \int_{\triangle V_t^O(x)}^{\infty} \left(p_t^O(w,x) + V_{t-1}^O(x-1)\right) f(w)dw\right]$$

$$= V_{t-1}^O(x) + \frac{q}{2}\left[\int_{\triangle V_t^O(x)}^{\infty} \left(w - \triangle V_t^O(x)\right) f(w)dw\right] \qquad x = 1,\ldots,X, t = 1,\ldots,T, \tag{6.1}$$

where, $\triangle V_t^O(x) = V_{t-1}^O(x) - V_{t-1}^O(x-1) \qquad x = 1,\ldots,X, t = 1,\ldots,T$, and

$$p_t^O(W,x) = \frac{\left(W + V_{t-1}^O(x) - V_{t-1}^O(x-1)\right)}{2} = \frac{(W + \triangle V_t^O(x))}{2} \qquad \begin{array}{l} x = 1,\ldots,X, \\ t = 1,\ldots,T, \end{array}$$

and boundary conditions $V_t^O(0) = 0 \quad t = 1,\ldots,T$, $V_0^O(x) = 0 \ x = 1,\ldots,X$.

Model 1 is not ex-post efficient, while Model 4 is. Below, we compare these two models (as well as the posted-price model, Model 2) to determine the cost of achieving ex-post efficiency for four distributions of the buyer's reservation value - uniform, exponential, normal, and triangular. Again, the results of the comparisons of this section are rather insensitive to change in the probability of customer arrival (q) and hence, we consider only two cases: (i) $q = 0.9$ and (ii) $q = 0.3$.

First, consider the uniform case - $W \sim U[0,1]$. It is easy to show analytically that the seller's optimal value functions are the same for Model 4 and Model 2. Thus, Figure 5.6 also compares Model 1 to Model 4.

Second, consider the exponential case - $W \sim exp(1)$. Figure 6.6 compares Model 1 to Model 4. Looking at Figures 6.2 and 6.6, we see that the ex-post efficient model outperforms the posted-price model in terms of the seller's optimal value functions in all states.

Finally, Figures 6.7 and 6.8 show that the cost of achieving ex-post efficiency is considerable when the buyer's reservation value is distributed normally or triangularly, respectively. In these two cases, Model 4 also results in lower optimal value functions in all states compared to Model 2, as observed from Figures 5.7, 6.3, 6.7 and 6.8.

Thus, utilizing an ex-post efficient model as opposed to the bargaining model proposed in chapter 5 can result in significant loss of expected revenue for the seller. At the same time, it results in the best aggregate value (refer §5.4) for the buyers, as the opportunity cost is a lower bound on the seller's choice of its reservation value. Finally, the ex-post efficient model outperforms the posted-price model in terms of the seller's optimal value function, in the

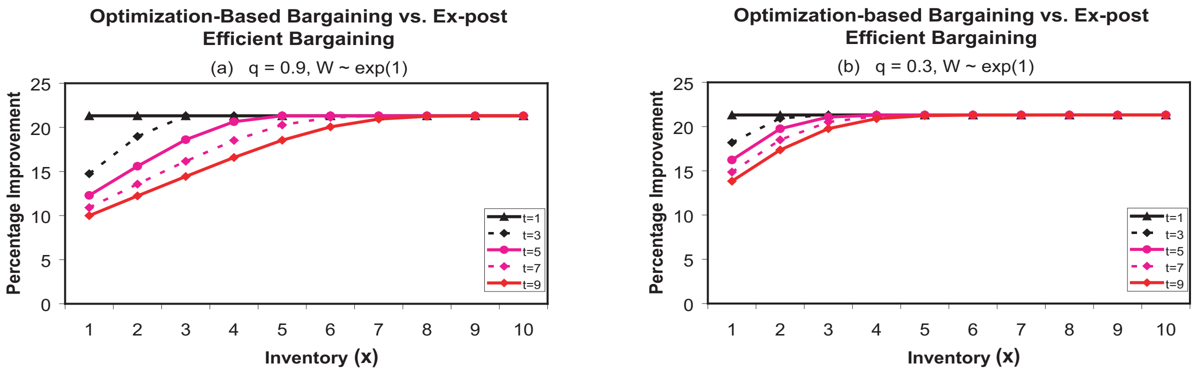

Figure 6.6: Exponential distribution - Percentage improvement of optimal value functions of the optimization-based bargaining model over those of ex-post efficient bargaining model.

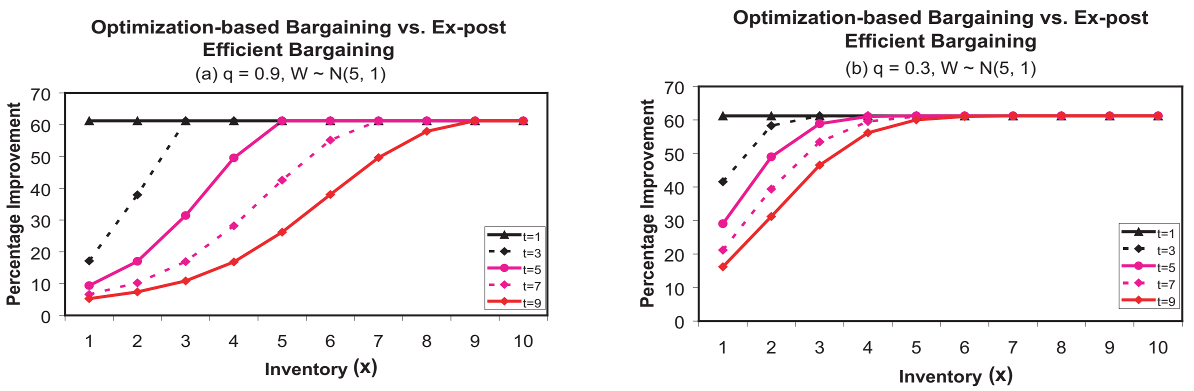

Figure 6.7: Normal distribution - Percentage improvement of optimal value functions of optimization-based bargaining model over those of ex-post efficient bargaining model.

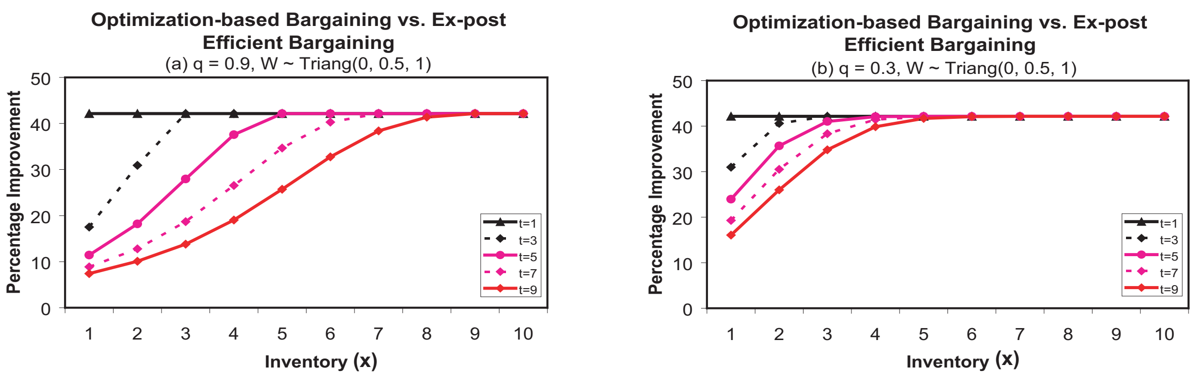

Figure 6.8: Triangular distribution - Percentage improvement of optimal value functions of Model 1 over Model 4.

uniform and exponential distribution cases but not in the normal and triangular distribution cases of the buyer's reservation value. Thus, it results in a Pareto improvement over Model 2 in the former two cases.

6.4 Short Summary

We present additional numerical experiments for our bargaining model presented in chapter 5. We numerically show that the seller's expected immediate revenue is also always greater in the bargaining model as compared to the posted-price model. We also discuss the concept of ex-post efficiency of a bargaining model and show that our bargaining model is not ex-post efficient. We numerically show that if the seller adopts an ex-post efficient bargaining model, the decrease in its expected revenue can be substantial compared to our bargaining model.

In the next chapter, we provide conclusions and directions for future work for all the chapters in this dissertation.

Chapter 7

Conclusion

7.1 Chapter 2 - Summary and Future Work

We show that the system of balance equations to obtain the limiting probability values of any ergodic irreducible Markov chain are equivalent to a system of balance inequalities. This result may be used to develop mathematical programming formulations for a constrained Markov decision process. The MBEDC algorithm is one exact and efficient solution technique to obtain the optimal non-randomized policy for any constrained Markov decision process. The CPLEX USER manual (chapter 3) recommends certain settings for its MIP solver to improve the performance for problems such as solved by the MBEDC algorithm. We tried all these recommendations and found that the following two improved the performance of the MBEDC algorithm considerably: (i) Setting the backtrack parameter for the MIP solver to its maximum value of 1, and (ii) recognizing that the constraints requiring the binary variables to sum to one are SOS constraints of type 1. With these settings, the MBEDC algorithm finds "good" integer feasible solutions quickly, but spends the major portion of the computation time in improving the best node value, i.e., the current best lower bound. This is because the MBEDC algorithm introduces some degeneracy into the formulation, i.e., multiple solutions with the same objective value are possible for the relaxed sub-problems when not all the binary variables are forced to be integers. CPLEX is forced to explore multiple nodes whose resulting children have the same objective value as the parents (the node from which they were obtained) in the branch and bound tree. Hence, the improvements in the best node in CPLEX are very slow. However, the MBEDC algorithm is the only efficient technique currently to obtain the optimal non-randomized policy for constrained MDPs. If any future study is undertaken to improve the performance of the MBEDC algorithm or to find an algorithm with better performance, this dissertation provides details of studies already performed with this goal in mind. Hence, any future efforts can start where this dissertation stops. For example, formulations (F1), (F2), and (F3) are studied for this reason, however, they are not efficient for the CDOS problem. In the future, we plan to study the computational time required for Formulation (F1), the BEDP algorithm, to solve the multi-item spare parts problem introduced in chapter 4. The BEDP algorithm does not introduce degeneracy into the formulation because of the mostly block structure of its constraint co-efficient matrix, but its performance needs to be tested. Finally, the flowchart in Figure 7.1 details how to employ the MBEDC algorithm for any constrained MDP problem with p parameters. In particular for the CDOS problem, $p = 4$ (x, y, n_1, n_2), and $m = 2$ (x, y), and PI, VI and LP refer to policy iteration, value iteration and linear programming, respectively.

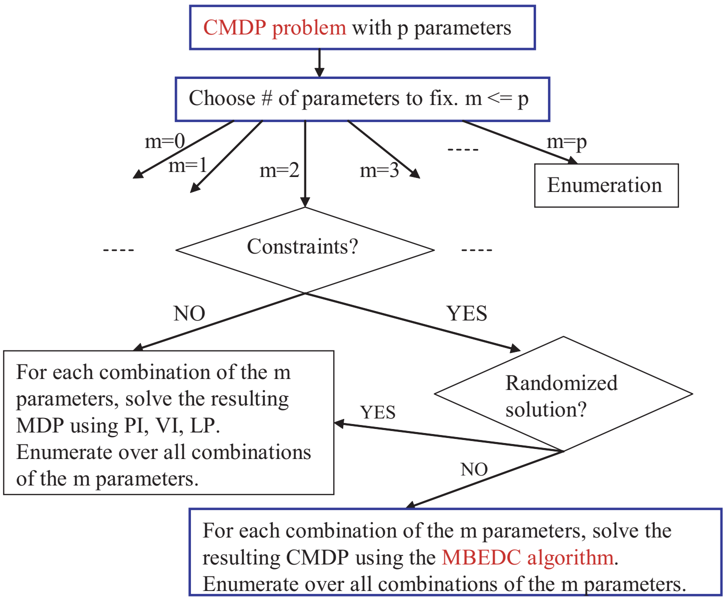

Figure 7.1: MBEDC strategy. PI = policy iteration, VI = value iteration, LP = linear programming.

7.2 Chapters 3 and 4 - Summary and Future Work

The CDOS problem is a very relevant problem not only in call centers but in many other service industries such as flexible Internet servers (Akamai Technologies, www.akamai.com), restaurants, banks, grocery stores, etc. The key feature all of these businesses share is that service level goals are important and hence must be included in their operational model. Traditionally, these goals have been included as opportunity costs, but such costs are difficult to estimate, whereas modeling the goals as constraints is straightforward.

We are the first to utilize Integer Programming techniques to solve the CDOS problem with explicit service level constraints for the optimal non-randomized policy. As such we provide a bridge between Integer Programming and constrained MDPs, demonstrating how the former can be used to efficiently solve the latter. Our MBEDC algorithm is an efficient, exact solution method for the CDOS problem and its relative performance improves over enumeration as the problem size increases. In general, the MBEDC algorithm can be used in two ways: (i) Direct implementation to find optimal solutions in real time for real world problems, or (ii) to benchmark heuristics (instead of lower/upper bounds) for real world problems.

In the CDOS problem, we find that the opportunity costs have a tremendous effect on the optimal number of temporary operators (cf. Section 3.4.2), and hence it is crucial they are accurately estimated. In our experiments we find that it is optimal to use at least one temporary operator for all ranges of the service level targets. Moreover, the value of flexibility provided by temporary operators in the case of very stringent or very relaxed service constraints changed within a range of 2-8%. If hiring costs dominate, each temporary operator is effective at reducing cost. On the other hand, if the opportunity cost dominate, then only one or possibly two temporary operators are effective at reducing costs. Moreover, we find that the call center manager should invest no (or negligible) resources compared to the hiring and opportunity costs in trying to determine the real time arrival rate. The main concern should be to determine the correct staffing levels of the permanent and temporary operators and possibly just a single threshold value for the number in queue at which the temporary operators are asked to provide service.

Satisfying a service level constraint (beyond some minimal threshold level) imposes costs on a service provider. Temporary operators can be an important tool in reducing these costs, as they provide finer staffing control for the call center manager. While we focus on the probability of no delay constraint and the number in queue constraint, the economic cost of satisfying any service level goal (constraint) can be obtained similarly, by comparing the optimal objective values of the CDOS problem with and without that particular constraint, respectively.

In the future we also plan to study: (i) CDOS problem with multiple customer classes, and (ii) CDOS problem where the manager only needs to satisfy l of the L service level goals using Disjunctive Programming techniques (Balas, 1998). In the infinite buffer CDOS problem with multiple customer classes, applying the matrix analytic method is more difficult as the resulting Markov chains will be infinite in multiple dimensions. However, an approximation scheme called dimensionality reduction has been proposed for such problems in Harchol-Balter et al. (2003).

In chapter 4 we study two applications of constrained MDPs in the literature. For the ADT threshold policy problem we show that percentage decrease in the mean response time is small from adopting the optimized general threshold policy as opposed to the optimized ADT policy. For the spare parts problem, we provide the first exact solution method. However, we can solve only up to seven items easily. Ours is the first study that obtains the optimal non-randomized policy for both of these problems by using the MBEDC algorithm.

7.3 Chapters 5 and 6 - Summary and Future Work

In chapter 5, we propose a framework to incorporate price-bargaining into revenue-management problems. Our results provide a theoretical support to the existing practice of bargaining in B2B settings. First, we show that there is significant benefit in expected revenue if the seller adopts a dynamic rather than a myopic price-bargaining approach. Second, we establish that the seller is always (weakly) better off by bargaining, rather than posting prices, and is often significantly better off. Third, we indicate how the seller can improve its expected revenue by price-bargaining compared to posted-price without decreasing the buyers' expected value. The framework proposed in this thesis and the results obtained motivate the following extensions.

- **Multiple-issue bargaining.** The seller and the buyer negotiate multiple issues such as price, quantity, delivery time, multiple-product prices and quantities, etc., in a revenue-management setting. In particular, if the seller negotiates both the price and quantity with the buyer, then the seller must determine its optimal reservation value as a function of quantity. The bargaining outcome can still be modeled using a Nash bargaining solution, with the utility functions of the two parties suitably modified to incorporate both price and quantity considerations.
- **Strategic customers.** We have assumed that the buyers are myopic, i.e., the buyer's reservation value does not depend on the transactional mechanism (bargaining or posted-pricing). An important challenge is to model the behavior of buyers who behave strategically, for example as in Talluri and van Ryzin (2004a) for the posted-price case. In this case, the buyers may optimize their reservation values, which may become different from their true reservation values.
- **Competition.** The only way to consider competition on the seller's side in the current model is to predict its effect on the buyer's reservation-value random variable. A more sophisticated model that considers competition should be developed as in Netessine and

Shumsky (2001) for the posted-price case. A relevant question in the presence of competing sellers is to determine the equilibrium behavior of each seller. For example, if two sellers are competing, will they both decide to adopt a bargaining policy, or will they both adopt a posted-price policy, or will one adopt a posted-price policy and the other a bargaining policy?

In chapter 6, we reinforce the results from chapter 5 by conducting additional numerical experiments using two different distributions for the buyer's reservation value - exponential and triangular. We also show that the expected immediate reward for the seller is always greater in the bargaining model as compared to the posted-price model for all four distributions of the buyer's reservation value. We explain this improvement by studying the two factors, (i) change in probability of sale, and (ii) change in the expected sale price conditional on a sale, and show that the contributions of both of these factors are positive. Finally, if the seller adopts an ex-post efficient bargaining policy, then its expected revenue decreases significantly compared to the bargaining model. However, the seller may still have higher expected revenue than the posted-price case in the uniform and exponential distribution cases for the buyer's reservation value.

Appendix A

Appendix for Chapters 5 and 6

A.1 Proof of Properties of Model 1

Proof. We prove the properties of Model 1 listed in §5.2.

(i) For any state (x, t) (from (5.2)),

$$V_t^N(x) = V_{t-1}^N(x) + q \ \mathrm{max}_{u \geq 0} \left[\int_u^\infty \left(\frac{w+u}{2} - \triangle V_t^N(x) \right) f(w) \ dw \right] \geq V_{t-1}^N(x)$$

(ii) The proof is identical to the proof developed in Zhao and Zheng (2000), and we only need to replace prices in the proof (p) by reservation values (u).

(iii) Proof is by mathematical induction. $V_0^N(x)$ is trivially increasing in x. Assume $V_{t-1}^N(x)$ is increasing in x. Also from (ii), $V_t^N(x)$ is concave in x for all $t \in \{1, \ldots, T\}$. Hence, $\triangle V_t^N(x) \leq \triangle V_t^N(x-1)$ in any state (x, t). From (5.2),

$$V_t^N(x) - V_t^N(x-1) = V_{t-1}^N(x) + q \ \mathrm{max}_{u \geq 0} \int_u^\infty \left(\frac{w+u}{2} - \triangle V_t^N(x) \right) f(w) \ dw$$

$$- V_{t-1}^N(x-1) - q \ \mathrm{max}_{v \geq 0} \int_v^\infty \left(\frac{w+v}{2} - \triangle V_t^N(x-1) \right)$$

$$\geq 0 + q h_t(x) \quad (V_{t-1}^N(x) \geq V_{t-1}^N(x-1) \text{ from induction hypothesis})$$

To complete the proof note that,

$$h_t(x) = \mathrm{max}_{u \geq 0} \int_u^\infty \left(\frac{w+u}{2} - \triangle V_t^N(x) \right) f(w) dw - \mathrm{max}_{v \geq 0} \int_v^\infty \left(\frac{w+v}{2} - \triangle V_t^N(x-1) \right) f(w) \ dw$$

$$\geq \max_{u\geq 0}\int_u^\infty\left(\frac{w+u}{2}-\triangle V_t^N(x)\right)f(w)dw-\max_{v\geq 0}\int_v^\infty\left(\frac{w+v}{2}-\triangle V_t^N(x)\right)f(w)\ dw\ =0$$

(iv) For any given state (x,t), u^* must satisfy the following first order condition (assuming differentiability of $F(u)$): $\frac{\partial}{\partial u}\left[V_{t-1}^N(x)\ F(u)+\int_u^\infty\left(\frac{w+u}{2}+V_{t-1}^N(x-1)\right)f(w)\ dw\right]=0$. By Leibnitz's rule, this implies that $u^*=\triangle V_t^N(x)+\frac{1}{2}\frac{\bar{F}(u^*)}{f(u^*)}$ provided that $f(u^*)\neq 0$ (here $\bar{F}(\cdot):=1-F(\cdot)$). Note that $\frac{\bar{F}(u^*)}{f(u^*)}$ is the inverse of the hazard rate of the buyer's reservation value random variable at u^*. If $f(u^*)=0$, optimality of u^* implies that $\bar{F}(u^*)=0$, i.e., $u^*\geq w^{max}\geq\triangle V_t^N(x)$, where $w^{max}:=inf\{w:\bar{F}(w)=0\}$. Hence, $u_t^*(x)\geq\triangle V_t^N(x)$ in any state (x,t).

(v) We will use the following theorem from Topkis (1978):

Theorem A.1.1. *If (a) g is supermodular (submodular) on the set of feasible states and actions, C, (b) $A(\cdot)$ is ascending on the set of feasible states, and (c) $A^*(s)$ is nonempty for every state s, then (d) $A^*(\cdot)$ is ascending (descending) on the set of feasible states.*

(a) Let $g_t(u,x)=\int_u^\infty\left(\frac{w+u}{2}-\triangle V_t^N(x)\right)f(w)\ dw$. Then, $u_t^*(x)=\arg\max_{u\geq 0} g_t(u,x)$. We drop the subscript t from $g_t(\cdot)$ and $\triangle V_t^N(x)$ and denote the state by x. Lemma A.1.1 (a), below, gives that $g(u,x)$ is submodular on the set of feasible u and x. Let $A(x)$ denote the set of admissible values for u ($\{u:u\geq 0\}$) when the inventory is x. $A(\cdot)$, called the set function (Topkis, 1978), is same for all x and, hence, is trivially ascending (Topkis, 1978) in x. Let $A^*(x)$ denote the set of optimal actions for state x. It can be easily shown that all the conditions of Theorem A.1.1 are met, and hence $u_t^*(x)$ is decreasing in x for any $t\in\{1,\ldots,T\}$.

(b) Let $h_x(u,t)=\int_u^\infty\left(\frac{w+u}{2}-\triangle V_t^N(x)\right)f(w)\ dw$. Then, $u_t^*(x)=\arg\max_{u\geq 0} h_x(u,t)$. We drop the subscript x from $h_t(\cdot)$ and $\triangle V_t^N$ and denote the state by t. Lemma A.1.1 (b), below, gives that $h(u,t)$ is supermodular on the set of feasible u and t. Let $A(t)$ denote the set of admissible values for u ($\{u:u\geq 0\}$) when the time to go is t. $A(\cdot)$, called the set function, is same for all t and, hence, is trivially ascending in t (Topkis, 1978). Let $A^*(t)$ denote the set of optimal actions for state t. It can be easily shown that all the conditions of Theorem A.1.1 are met and hence, $u_t^*(x)$ is increasing in t for any $x\in\{1,\ldots,X\}$.

Lemma A.1.1. *(a) $g(u,x)$ is submodular, i.e., for any u,u',x,x' (in the domain of g) such that $\bar{u}=u\vee u'$, $\bar{x}=x\vee x'$, $\underline{u}=u\wedge u'$, and $\underline{x}=x\wedge x'$, we have $g(u,x)+g(u',x')\geq g(\underline{u},\underline{x})+g(\bar{u},\bar{x})$. (b) $h(u,t)$ is supermodular, i.e., for any u,u',t,t' (in the domain of h) such that $\bar{u}=u\vee u'$, $\bar{t}=t\vee t'$, $\underline{u}=u\wedge u'$, and $\underline{t}=t\wedge t'$, we have $h(u,t)+h(u',t')\leq h(\underline{u},\underline{t})+h(\bar{u},\bar{t})$.*

Pf : (a) Assume without loss of generality that $u\leq u'$. Then

$$I = g(u,x) + g(u',x') - g(\underline{u},\underline{x}) - g(\bar{u},\bar{x}) = g(u,x) + g(u',x') - g(u,\underline{x}) - g(u',\bar{x})$$

$$= -\triangle V^N(x)\bar{F}(u) - \triangle V^N(x')\bar{F}(u') + \triangle V^N(\underline{x})\bar{F}(u) + \triangle V^N(\bar{x})\bar{F}(u')$$

$$= \bar{F}(u)\left[\triangle V^N(\underline{x}) - \triangle V^N(x)\right] - \bar{F}(u')\left[\triangle V^N(x') - \triangle V^N(\bar{x})\right]$$

If $x \le x'$, then $I = 0$. If $x > x'$ then, $I = [\bar{F}(u) - \bar{F}(u')]\left[\triangle V^N(x') - \triangle V^N(x)\right] \ge 0$. (Since $u \le u'$ and $x' < x$.) Thus, $g(u,x)$ is submodular.

(b) Assume without loss of generality that $u \le u'$. Then

$$I = h(u,t) + h(u',t') - h(\underline{u},\underline{t}) - h(\bar{u},\bar{t}) = h(u,t) + h(u',t') - h(u,\underline{t}) - h(u',\bar{t})$$

$$= -\triangle V_t^N \bar{F}(u) - \triangle V_{t'}^N \bar{F}(u') + \triangle V_{\underline{t}}^N \bar{F}(u) + \triangle V_{\bar{t}}^N \bar{F}(u')$$

$$= \bar{F}(u)\left[\triangle V_{\underline{t}}^N - \triangle V_t^N\right] - \bar{F}(u')\left[\triangle V_{t'}^N - \triangle V_{\bar{t}}^N\right]$$

If $t \le t'$, then $I = 0$. If $t > t'$ then, $I = [\bar{F}(u) - \bar{F}(u')]\left[\triangle V_{t'}^N - \triangle V_t^N\right] \le 0$. (Since $u \le u'$ and using Proposition A.1.2, below.) Hence, $h(u,t)$ is supermodular. ∎

Lemma A.1.2. *For Model 1, $\triangle V_{t+1}^N(x) \ge \triangle V_t^N(x)$ in any state (x,t).*

Pf : Note that

$$\triangle V_{t+1}^N(x) = V_t^N(x) - V_t^N(x-1) = \triangle V_t^N(x) + q\ I$$

where,

$$I = \max_{v\ge 0}\left[\int_v^\infty \left(\frac{w+v}{2} - \triangle V_t^N(x)\right) f(w)dw\right] - \max_{u\ge 0}\left[\int_u^\infty \left(\frac{w+u}{2} - \triangle V_t^N(x-1)\right) f(w)dw\right]$$

$$\ge \max_{v\ge 0}\left[\int_v^\infty \left(\frac{w+v}{2} - \triangle V_t^N(x)\right) f(w)dw\right] - \max_{u\ge 0}\left[\int_u^\infty \left(\frac{w+u}{2} - \triangle V_t^N(x)\right) f(w)dw\right]$$

$= 0$. (The first inequality is true since $\triangle V_t^N(x-1) \ge \triangle V_t^N(x)$ from the concavity of $V_t^N(x)$.)

Thus, $\triangle V_{t+1}^N(x) \ge \triangle V_t^N(x)$ for any state (x,t). ∎ □

A.2 Structural Properties of Optimal Value Functions of Model 4

The structural properties of the seller's optimal value functions from Model 4 are provided below.

Proposition A.2.1. *(Properties of the seller's optimal value functions from Model 4:)*
(i) $V_t^O(x)$ *is increasing in* t *for any* x *(*$t = T$ *at start and* $t = 0$ *at the end of the horizon).*
(ii) $V_t^O(x)$ *is concave in* x *for any* t*.*
(iii) $V_t^O(x)$ *is increasing in* x *for any* t*.*

Proof. (i) For any state (x, t), we obtain from (6.1),

$$V_t^O(x) = V_{t-1}^O(x) + \frac{q}{2}\left[\int_{\triangle V_t^O(x)}^{\infty} \left(w - \triangle V_t^O(x)\right) f(w)dw\right] \geq V_{t-1}^O(x)$$

(ii) We use mathematical induction for the proof. The optimal value function equation (**??**) from Model 4 can also be written as:

$$V_t^O(x) = (1-q)V_{t-1}^O(x) + q\ E_W\left[max\left[p^N(w,x,t) + V_{t-1}^O(x-1), V_{t-1}^O(x)\right]\right] \quad \begin{matrix} x = 1,\ldots,X \\ t = 1,\ldots,T \end{matrix} \tag{A.1}$$

Let $g(w,x) = max\left[p^N(w,x,t) + V_{t-1}^O(x-1), V_{t-1}^O(x)\right]$. Then from (A.1) we have:

$$V_t^O(x) = (1-q)V_{t-1}^O(x) + q\ E_W\ [g(w,x)] \quad x = 1,\ldots,X \quad t = 1,\ldots,T \tag{A.2}$$

$V_0^O(x)$ is trivially concave in x. Assume $V_{t-1}^O(x)$ is concave in x ($t \leq T$). From Lemma A.2.1, $g(w,x)$ is concave in x for any w, and hence $E_W[g(w,x)]$ is also concave in x. Then from (A.2), $V_t^O(x)$ is concave in x. Thus by mathematical induction, $V_t^O(x)$ is concave in x for any t.

Lemma A.2.1. $g(w,x)$ *is concave in* x *for any* w*.*

Pf :

$$g(w,x) = max\left[p^N(w,x,t) + V_{t-1}^O(x-1), V_{t-1}^O(x)\right]$$

$$= max\left[\frac{w}{2} + \frac{V_{t-1}^O(x)}{2} + \frac{V_{t-1}^O(x-1)}{2}, V_{t-1}^O(x)\right]$$

Let $\widehat{x} = max\{x : \triangle V_t^O(x) \geq w\}$. Since $\triangle V_t^O(x)$ is decreasing in x, $\triangle V_t^O(x) \geq w$ for all $x \leq \widehat{x}$ and $\triangle V_t^O(x) < w$ for all $x > \widehat{x}$.

Case 1: $x < \widehat{x}$

$\Rightarrow g(w,x) = V_{t-1}^O(x)$

This is concave and increasing in x. Thus g(w,x) is concave and increasing in x in the interval $0 \leq x < \widehat{x}$.

Case 2: $x > \widehat{x}$

$\Rightarrow g(w,x) = \frac{w}{2} + \frac{V_{t-1}^O(x)}{2} + \frac{V_{t-1}^O(x-1)}{2}$

This is concave and increasing in x. Thus g(w,x) is concave and increasing in x in the interval $x > \widehat{x}$.

Thus, we need to show that $g(w,x)$ is concave at $x = \widehat{x}$ to prove that $g(w,x)$ is concave in x. Let $I = (g(w,\widehat{x}+1) - g(w,\widehat{x}))$ - $(g(w,\widehat{x}) - g(w,\widehat{x}-1))$

$= \left(\frac{w}{2} + \frac{V_{t-1}^O(\widehat{x}+1)}{2} + \frac{V_{t-1}^O(\widehat{x})}{2} - V_{t-1}^O(\widehat{x})\right) - \left(V_{t-1}^O(\widehat{x}) - V_{t-1}^O(\widehat{x}-1)\right)$

Note that $w = V(\widehat{(x)}) - V(\widehat{x}-1)$. Thus,

$$I = \frac{V_{t-1}^O(\widehat{x})}{2} - \frac{V_{t-1}^O(\widehat{x}-1)}{2} + \frac{V_{t-1}^O(\widehat{x}+1)}{2} + \frac{V_{t-1}^O(\widehat{x})}{2} - 2V_{t-1}^O(\widehat{x}) + V_{t-1}^O(\widehat{x}-1)$$

$$= \frac{V_{t-1}^O(\widehat{x}+1)}{2} + \frac{V_{t-1}^O(\widehat{x}-1)}{2} - V_{t-1}^O(\widehat{x})$$

But $V_{t-1}^O(x)$ is concave in x. Thus, $I \leq 0$ and $g(w,x)$ is concave in x at $x = \widehat{x}$. Hence, $g(w,x)$ is concave in x.

■

(iii) From Model 4, the optimal value function equation is

$$V_t^O(x) = V_{t-1}^O(x) + \frac{q}{2}\left[\int_{\triangle V_t^O(x)}^{\infty} \left(w - \triangle V_t^O(x)\right) f(w)dw\right] \quad x = 1, \ldots, X \quad t = 1, \ldots, T$$

We show that the right hand side of the above equation is increasing in x by using mathematical induction. $V_0^O(x)$ is trivially increasing in x. Assume that $V_{t-1}^O(x)$ is increasing in x $(t \leq T)$. Also $\triangle V_t^O(x)$ is a decreasing function of x since $V_{t-1}^O(x)$ is a concave function of x (from part (ii) above). Thus, the right hand side of the above equation is increasing in x. □

Bibliography

Ahn, H. S., I. Duenyas and R. Q. Zhang. 2004. Optimal control of a flexible server. *Advances in Applied Probability* **36** 139-170.

Akamai Technologies, Inc. 2003. Akamai Technologies: Technical overview. *Akamai Technologies White Paper*, www.akamai.com.

Altman, A. 1999. *Constrained Markov Decision Processes.* Chapman & Hall/CRC, Boca Raton, Florida.

Andrews, B. H. and H. L. Parsons. 1993. Establishing telephone agent staffing levels through economic optimization. *Interfaces* **23** 14-20.

Balas, E. 1998. Disjunctive Programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics* **89**(1-3) 3-44.

Bertsekas, D. 1995. *Dynamic Programming and Optimal Control, Volume Two*, Athena Scientific, Belmont Massachusetts.

Bitran, G. R. and R. Caldentey. 2003. An overview of pricing models for revenue management. *Manufacturing & Service Operations Management*, **5** 203-229.

Bitran, G. and S. Mondschein. 1997. Periodic pricing of seasonal products in retailing. *Management Science*, **43** 64-79.

Boyd, A. and I. C. Bilegan. 2003. Revenue management and E-commerce. *Management Science* **49** 1363-1386.

Brigandi, A. J., D.R. Dargon, M.J. Sheehan and T. Spencer III. 1994. AT&T's call processing simulator (CAPS) operational design for inbound call centers. *Interfaces* **2**4 6-28.

Corfman, K. P. and S. Gupta. 1993. Mathematical models for group choice and negotiations. In J. Eliahberg and G.L. Lilien (eds.), *Marketing*, Vol. 5 of Handbook in OR and MS, 83-142, Elsevier, Amsterdam, The Netherlands.

Cox, D. R. and W. L. Smith. 1971. *Queues*, Kluwer Academic Publishers, Boston, MA.

Deniz, B., I. Karaesmen and A. Scheller-Wolf. 2005. Managing inventories of perishable goods. *Working Paper*, Tepper School of Business, Carnegie Mellon University.

Elmaghraby, W. and P. Keskinocak. 2003. Dynamic pricing in the presence of inventory considerations: Research overview, current practices and future directions. *Management Science* **49** 1287-1309.

Gallego, G. and G. van Ryzin. 1994. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science* **40** 999-1020.

Gans, N., G. Koole and A. Mandelbaum. 2003. Telephone call centers: Tutorial, review and research prospects. *Invited review paper by Manufacturing and Service Operations Management (M&SOM)*, **5** 79-141.

Geoffrion, A. M. and R. Krishnan. Prospects for operations research in the e-business era. *Interfaces* **31**(2) 6-36.

Hall R. 1991. *Queueing Methods for Services and Manufacturing*, Prentice Hall, Englewood Cliffs, New Jersey.

Harchol-Balter, M., C. Li, T. Osogami, A. Scheller-Wolf and M. Squillante. 2003. Task assignment with cycle stealing under central queue. *23rd International Conference on Distributed Computing Systems (ICDCS '03)* , Providence, RI.

Ingolfsson, A., M. A. Haque and A. Umnikov. 2002. Accounting for time-varying queueing effects in workforce scheduling. *Eur. J. Oper. Res.* **139** 585-597.

Jennings, O., A. Mandelbaum, W. Massey and W. Whitt. 1996. Server staffing to meet time-varying demand. *Management Science* **42** 1383-1394.

Jongbloed, G. and G. M. Koole. 2001. Managing uncertainty in call centers using Poisson mixtures. *Appl. Stochastic Models in Bus. Indust.* **17** 307-318.

Lippman, S. A. and K.F. McCardle. 2004. Sex, lies and the Hillblom estate: A decision analysis. *Decision Analysis* **1** 149-166.

Lucantoni, D. 1993. The BMAP/G/1 queue: A tutorial. *Performance/SIGMETRICS Tutorials.* 330-358.

Meyn, S. 2001. Sequencing and routing in multiclass queueing networks part i: Feedback regulation. *SIAM Journal on Control Optimization* **40** 741-776.

Muthoo, A. 1999. *Bargaining Theory with Applications.* Cambridge University Press, Cambridge, UK.

Nain, P. and K. W. Ross. 1986. Optimal priority with hard constraints. *IEEE Trans. Automatic Control* **31** 883-888.

Nain, P. and R. Núñez-Queija. 2001. An M/M/1 queue in a semi-Markovian environment. *Proceedings of the ACM SIGMETRICS / Performance 2001 Conference*, Cambridge, Massachusetts, 268-278.

Nash, J. F. 1950. The bargaining problem. *Econometrica* **18** 155-162.

Nemhauser, G. and L. Wolsey. 1988. *Integer and Combinatorial Optimization*, John Wiley &

Sons, New York, New York.

Netessine, S. and R. A. Shumsky. 2001. Revenue management games. Simon School of Business Working Paper No. OP 00-04, University of Rochester, NY.

Neuts, M. 1981. *Matrix-Geometric Solutions in Stochastic Models.* Johns Hopkins University Press, Baltimore, Maryland.

Osogami, T. and M. Harchol-Balter. 2003. Necessary and sufficient conditions for representing general distributions by Coxians. *13th International Conference on Modeling Techniques and Tools for Computer Performance Evaluation*, Urbana, Illinois 182-199.

Osogami, T., M. Harchol-Balter and A. Scheller-Wolf. 2005. Robustness of Threshold Policies for Beneficiary-Donor Model. *The Seventh Workshop on Mathematical Performance Modeling and Analysis*, Banff, Canada; *Performance Evaluation Review* **33** 36-38.

Porteus, E. 2002. *Foundations of Stochastic Inventory Theory*, Stanford University Press, Stanford, California.

Puterman, M. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley and Sons, New York, New York.

Rangaswamy, A. and G. R. Shell. 1997. Using computers to realize joint gains in negotiations: Toward an "Electronic bargaining table". *Management Science* **43** 1147-1163.

Ross, S. M. 1997. *Introduction to probability models*, Academic Press, San Diego, California.

Rubinstein, A. 1982. A perfect equilibrium in a bargaining model. *Econometrica* **50** 97-109.

Secomandi, N. 2006. Research prospects on revenue management for business-to-business negotiations. Tepper Working Paper 2004-04, Carnegie Mellon University, PA.

Sherbrooke, C. C. 2004. *Optimal inventory modeling of systems*, Kluwer, Boston, MA.

Shwartz, A. and A. M. Makowski. 1990. Comparing policies in Markov decision processes: Mandl's lemma revisited. *Math. of Operations Research* **15** 155-174.

Squillante, M. S., C. H. Xia and L. Zhang. 2001. Optimal scheduling in queuing network models of high-volume commercial web sites. *Performance Evaluation* **47** 223-242.

Talluri, K. T. and G. van Ryzin. 2004a. Revenue management under a general discrete choice model of consumer behavior. *Management Science* **50** 15-33.

Talluri, K. T. and G. van Ryzin. 2004b. *The Theory and Practice of Revenue Management.* Kluwer Academic Publishers, Boston, MA.

Terwiesch, C., S. Savin and I.-H. Hann. 2005. Online haggling at a name-your-own-price retailer: Theory and application. *Management Science* **51** 339-351.

Topkis, D. M. 1978. Minimizing a submodular function on a lattice. *Operations Research* **26** 305-321.

Veeraraghavan, S. and A. Scheller-Wolf. 2006. Now or Later: A simple policy for effective dual sourcing in a capacitated environment. *Operations Research*, to appear.

Whitt, W. 1999. Dynamic staffing in a telephone call center aiming to immediately answer all calls. *Operations Resarch Letters* **24** 205-212.

Williams, R. J. 2000. On dynamic scheduling of a parallel server system with complete resource pooling. In D. R. McDonald and S. R. E. Turner, editors, *Analysis of Communication Networks: Call Centers, Traffic and Performance*, American Mathematical Society.

Wong, H., G. J. Van Houtum, D. Cattrysse and D. Van Oudheusden. 2006. Multi-item spare parts systems with lateral transshipments and waiting time constraints. *European Journal of Operational Research* **171** 1071-1093.

Yoo, J. 1996. *Queueing models for staffing service operations*, Ph.D. dissertation, University of Maryland, College Park, MD.

Zhao, W. and Y. S. Zheng. 2000. Optimal dynamic pricing for perishable assets with nonhomogeneous demand. *Management Science* **46** 375-388.