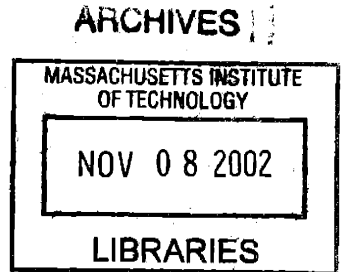


# Service Delivery and Learning in Automated Interfaces

by  
Paulo Rocha e Oliveira  
A.B., Princeton University, 1996

Submitted to the Sloan School of Management  
in partial fulfillment for the degree of  
Doctor of Philosophy in Management



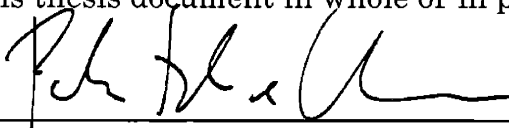
at the

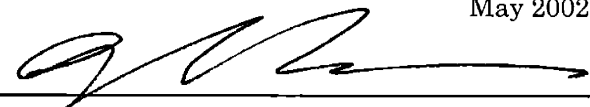
MASSACHUSETTS INSTITUTE OF TECHNOLOGY


September 2002

© Massachusetts Institute of Technology 2002


The author hereby grants to Massachusetts Institute of  
Technology permission to reproduce and to distribute copies of  
this thesis document in whole or in part.

Signature of Author:   
Sloan School of Management  
May 2002

Certified by:   
Gabriel R. Bitran  
Deputy Dean; Nippon Telephone and Telegraph Professor of Management  
Thesis Supervisor

Certified by:   
Dan Ariely  
Assistant Professor of Marketing  
Thesis Supervisor

Certified by:   
René Caldentey  
Assistant Professor of Operations Management – New York University

Accepted by:   
Birger Wernerfelt  
Professor of Management Science  
Chair of the Doctoral Program

1. The first part of the document  
 2. discusses the general principles  
 3. of the proposed system.  
 4. It is intended to provide a  
 5. clear and concise overview  
 6. of the key components and  
 7. objectives of the project.



# **Service Delivery and Learning in Automated Interfaces**

by

Paulo Rocha e Oliveira

## **Abstract**

This dissertation analyzes the strategic implications of customization policies available to companies that must simultaneously provide service and learn about their customers through automated interfaces. The first part of the dissertation lays out the theoretical framework within which the analysis is carried. The second part addresses whether companies should use Internet-based customization tools to design service encounters that maximize customers' utility in the present or explore customers' tastes to provide more value in the future. Good customization policies must quantify the value of knowledge so as to adequately balance the expected revenue of present and future interactions. Such policies can be obtained by analyzing the customization decision problem within the framework of dynamic programming. Interpretation of the service design policies enhances the current understanding of the mechanisms connecting service customization, value creation, and customer lifetime value. This leads to insights into the nature of the relationship between learning, loyalty, and long-term profitability in service industries. The final part of the dissertation considers situations where companies have the ability to acquire information by other means in addition to observing interactions with customers. In information-intensive industries, investments in customer retention often take the form of paying customers to answer questionnaires, or somehow acquiring information about the customers' preferences. The value of customers is convex as a function of knowledge. This means that the more firms know about a customer, the more eager they should be to learn even more. However, the cost of obtaining information about customers increases as knowledge increases. Understanding the interactions between these two functions is fundamental to designing information acquisition policies. In the real world, investment in customer retention must often be balanced with investment in customer acquisition. Therefore, investment in learning about a current customer must depend not only on the current level of knowledge about that customer but also on properties of the population to which potential customers belong. The analysis concludes with the characterization of information acquisition policies for a number of different managerial settings.

## **Committee:**

Gabriel R. Bitran  
Deputy Dean; Nippon Telephone and Telegraph Professor of Management  
Thesis Supervisor

Dan Ariely  
Associate Professor of Marketing  
Thesis Supervisor

René Caldentey  
Assistant Professor of Operations Management – New York University



# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>Partially Observable Markov Decision Processes in Management Science</b>	<b>13</b>
2.1	Introduction . . . . .	13
2.2	Problem formulation . . . . .	15
2.2.1	Markov Decision Processes . . . . .	15
2.2.2	Partially Observable Markov Decision Processes . . . . .	17
2.3	Applications . . . . .	21
2.3.1	Applications in Management Science . . . . .	21
2.3.2	Applications in other fields . . . . .	22
2.4	Solution Methods . . . . .	25
2.4.1	Value Iteration . . . . .	25
2.4.2	General Properties of POMDPs . . . . .	27
2.4.3	Exact Solution Methods . . . . .	30
2.4.4	Approximate Solution Methods . . . . .	37
2.5	New Approximation Method . . . . .	39
2.5.1	Approximating the Convex Hull . . . . .	39
2.5.2	Numerical Studies . . . . .	41
2.6	Conclusions . . . . .	43
2.7	Appendix . . . . .	44
2.7.1	Parameter Values . . . . .	44
2.7.2	The Exploration/Exploitation tradeoff . . . . .	44
<b>3</b>	<b>Design to Learn: Customizing Services when the Future Matters</b>	<b>46</b>
3.1	Introduction . . . . .	46
3.2	Model Description . . . . .	49
3.2.1	Customer Behavior . . . . .	49
3.2.2	Dynamics of Customer-Company Interactions . . . . .	51
3.2.3	Optimization Problem . . . . .	54
3.2.4	Summary . . . . .	56
3.3	Properties of the Value Function . . . . .	57
3.3.1	Optimality Conditions . . . . .	57

3.3.2	Value Iteration Algorithm . . . . .	58
3.3.3	Finite Representation of the Value Function . . . . .	58
3.4	Exact Solution Methods . . . . .	62
3.5	Derivation of Control Policies . . . . .	63
3.5.1	Value Function Approximations . . . . .	63
3.5.2	Analyzing the Bounds . . . . .	68
3.6	Discussion . . . . .	70
3.6.1	Solving the Agent's Dilemma . . . . .	70
3.6.2	Learning and Loyalty . . . . .	70
3.6.3	The Value of Knowing the Customer . . . . .	71
3.7	Directions for Future Research . . . . .	72
3.8	Appendix . . . . .	73
3.8.1	Proof of Theorem 14 . . . . .	73
3.8.2	Proof of Proposition 17 . . . . .	75
3.8.3	Proof of Proposition 18 . . . . .	76
<b>4</b>	<b>The Costs and Benefits of Information Acquisition</b>	<b>79</b>
4.1	Introduction . . . . .	79
4.2	Literature Review . . . . .	80
4.3	The Value of Information . . . . .	82
4.4	To Provide Service or to Ask Questions? . . . . .	84
4.4.1	One-Stage Interactions . . . . .	85
4.4.2	Two-Stage Interactions . . . . .	86
4.5	Learning vs. Customer Acquisition . . . . .	87
4.5.1	Quantifying the value of learning . . . . .	88
4.5.2	The Resource Allocation Problem . . . . .	92
4.5.3	Sensitivity Analysis . . . . .	105
4.6	Discussion . . . . .	108

# List of Figures

1-1	The dual role of automated customer interfaces . . . . .	7
1-2	The Customer Sacrifice Gap (CSG) . . . . .	9
1-3	An example of permission email . . . . .	10
2-1	Sequence of events in a Markov decision process . . . . .	16
2-2	Sequence of events in a Partially Observable Markov Decision Process . . . . .	18
2-3	The machine inspection problem . . . . .	21
2-4	Drake's symmetric two-state Markov source monitored by a binary symmetric channel . . . . .	23
2-5	A set $A^n$ of vectors and their corresponding value function . . . . .	28
2-6	Generating sets $A^n$ from $A^{n-1}$ . . . . .	30
2-7	Algorithms for POMDPs. . . . .	31
2-8	Geometric interpretation of Eagle's linear program . . . . .	33
2-9	A plot of the value function $V_t^0 = \max_{\alpha \in A_t^0} b \cdot \alpha$ . . . . .	34
2-10	A plot of the value function $V_t^1 = \max_{\alpha \in A_t^1} b \cdot \alpha$ . . . . .	34
2-11	Reducing the complexity of the update rule . . . . .	40
2-12	Comparing actual sets $A^n$ with their approximations $\tilde{A}^n$ . . . . .	41
2-13	Empirical comparison of the myopic policy $V^M(b)$ and the approximate policy $V^{\tilde{A}}(b)$ . . . . .	42
2-14	Simulation Results. Columns are initial belief levels (priors). Rows are different trials (parameter values). Entries are $\frac{V^{\tilde{A}}(b) - V^M(b)}{V^M(b)}$ . . . . .	43
3-1	Probabilities of Accepting, Rejecting, and Leaving when offered a product with deterministic utility $u$ . . . . .	51
3-2	The dynamics of customer-company interactions . . . . .	52
3-3	Transition probabilities and payoffs for the Fully Observable Stochastic Shortest Path problem . . . . .	55
3-4	Transition probabilities and payoffs in the Partially Observable Stochastic Shortest Path problem . . . . .	56
3-5	A set $A^n$ of vectors and their corresponding value function . . . . .	59
3-6	Generating sets $A^n$ from $A^{n-1}$ . . . . .	61
3-7	Comparison of optimal and myopic customization policies . . . . .	63

3-8	Comparing the policies $\mu^{LL(1)}$ (limited learning) and $\mu^M$ (myopic) . . . . .	67
3-9	Possible “extreme” configurations of customer segments and products . . . . .	69
3-10	The value of customers as a function of how well the company knows them . . . . .	71
4-1	The value of asking a question . . . . .	83
4-2	One-stage decision process . . . . .	85
4-3	Decision trees for the two-stage decision process . . . . .	86
4-4	The value of customers as a function of knowledge . . . . .	88
4-5	Learning functions for different levels of initial knowledge . . . . .	90
4-6	Profit from making an investment in learning of value $m$ when the level of knowledge is $k$ . . . . .	93
4-7	Optimal level of investment as a function of knowledge . . . . .	97
4-8	Profit function, relative increase in customer value, and return on investment when investment is made optimally . . . . .	97
4-9	Optimal level of investment under constraints . . . . .	99
4-10	Understanding the discontinuity in $m^*(k)$ . . . . .	99
4-11	Profit as a function of investment under constrained resources . . . . .	100
4-12	Customer acquisition rate as a function of investment . . . . .	103
4-13	Profitability of different levels of investment in customer acquisition . . . . .	104
4-14	Value functions corresponding to different interaction strategies . . . . .	105
4-15	Optimal investment levels for two different value functions . . . . .	106
4-16	Profit given optimal investment for two different value functions . . . . .	107
4-17	Learning policies of different effectiveness . . . . .	108
4-18	Optimal investments and their corresponding profit for different learning functions . . . . .	109



# Chapter 1

## Introduction

The delivery of services through automated interfaces requires the implementation of interface design strategies that result in encounters in which the firm actively gathers information while providing service. The design of these strategies requires an understanding of the value customer information and a quantification of the risk involved in unsuccessful service encounters.

The design of the customer interface is a strategic issue that has received a significant amount of attention in the services management literature. Customer interfaces are the point at which customers meet companies. Figure 1-1, based on the frameworks developed in [12] and [13], show how both service delivery and information gathering take place at the interface. Companies have clear objectives that are expressed in mathematical terms such as profit maximization, workforce scheduling, logistics, and optimization. Customers, on the other hand, are human beings and thus experience services subjectively. Therefore, the successful design of interfaces requires understanding customer behavior (psychology) in mathematical terms and delivering services that will satisfy subjective needs.

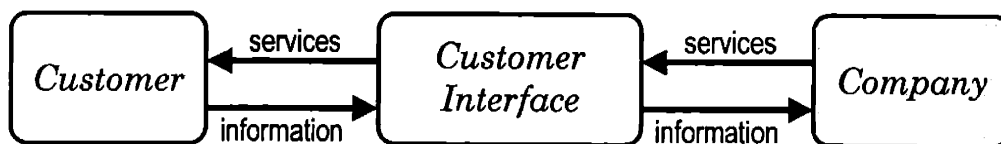


Figure 1-1: The dual role of automated customer interfaces

Up until very recently, the analysis of the service interface dealt with human interfaces. The problem with human interfaces is that a single employee can make a mistake and thereby taint the entire company's image. The quintessential case study from the early days of services operations management is Levitt's [54] study of McDonald's. This study exemplifies the philosophy of designing service delivery systems so robust that any employee with minimal intelligence and training would do exactly what was in the company's interest. Chase and Stewart [22] adapted the manufacturing concept of fail-safing and showed how service

delivery systems can have built-in devices to identify potential sources of failure and rectify mistakes before they happen. Another important issue in the design of human interfaces is the management of supply and demand. Employees are expensive which means that some customers will have to wait in order to receive service. Consequently, managing supply and demand and understanding the psychology of waiting have attracted the attention of a number of researchers such as Bitran and Mondschein [14] and Katz et al. [50].

When interfaces change from human-based to computer-based the main strategic decisions about their design also change. As the Internet came into existence, the traditional problems of associated with the control of customer interfaces seemed to go away. Providing consistent services is easy for computers that do exactly what they are programmed to do. Capacity constraints are not a problem if customers can go to a website instead of talking to a human customer service representative. However, an entire new set of problems has emerged. The link between the interface and the company became very strong, the link between the interface and the customer became weaker. Computers are not able to deliver determinants of quality such as courtesy, compassion or empathy [72] as effectively as humans. Urban et al. [93] describe how to increase the perception of a website's quality by introducing human cues into a computer interface so as to increase the customers' trust in the company. Ariely [4] reports that customers are much more likely to forgive a mistake when it is made by a human agent as opposed to a computer agent.

When the interface is automated, companies learn about customers by observing how they react with the interface. Therefore, the way in which the interface is configured can have a tremendous impact on the rate of learning. Customers react differently to different interfaces. Therefore, the problem of customizing services and interfaces is intimately linked to the way in which companies learn about their customers. In this way, interface design in the present has a direct effect on the company's future ability to provide customized services of high quality.

Customization is a prominent feature of the Internet. There is too much information on the Internet and people will not be able to find what they are looking for without the assistance of software programs. The Internet provides companies with many different ways in which they can customize their services. However, all these possibilities will be of no value to customers unless companies have the ability to provide customers with a small subset of options that closely match their preferences.

A large number of web-based companies correctly identified customization as an important business opportunity. Maes [64] describes how intelligent agents, the software programs that enable web-based customization, are becoming increasingly cheaper and easier to implement. Two important business changes have begun to take place. First, customization will be available to customers that could not previously afford it. Very few customers can afford the services of a human personal shopping agent. Any customer with a web connection can have an electronic personal shopping agent. Second, customization will begin to take place in industries where it was previously not feasible. Personalized radio stations and customized news services are rare (and for many people unheard of) in the brick-and-mortar

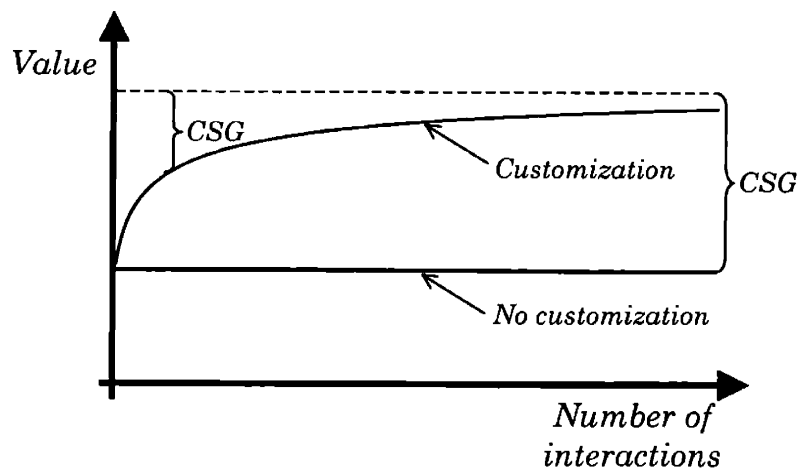


Figure 1-2: The Customer Sacrifice Gap (CSG)

world. These changes have already begun to have an impact on the competitive strategies in many industries.

Customization reduces what Gilmore and Pine [36] call the “customer sacrifice gap.” The customer sacrifice gap is the difference in value between the best possible service that could be offered to a customer (depicted by the dashed line in Figure 1-2) and the service that the customer experiences during a service encounter. When there is no customization, the customer sacrifice gap remains more or less constant over time. With customization, companies go up a learning curve as by observing the customers’ reactions to the services they receive. This means that customers are receiving increasingly higher value over time, which results in higher levels of loyalty and, consequently, profit. From a competitive stance, customization enables companies to erect entry barriers for their competitors. Customers that consistently receive customized services of high value know that their customer sacrifice gap will increase if they switch to a different company. The ability to provide quality to customers depends on knowledge of the customer.

Figure 1-3 gives an example of an automated web-based service. CNN quick news is an example of a permission email service. When customers are browsing CNN’s website they may notice a banner ad offering them the opportunity to receive email messages with customized news. If they click on this banner, they are taken to a page where they will enter their contact and demographic information and answer a few questions about the type of news they are interested in receiving. The customer will then receive email messages at a certain pre-determined frequency.

The customer who receives the email in Figure 1-3 will read a few headlines and click on the “Full story” link, in which case they will be immediately taken to the CNN website where they can read more about the story. If they are not interested in that particular piece of news, they can simply delete the message or ignore it. If, however, they find that the



Figure 1-3: An example of permission email

customized email is utterly uninteresting and the CNN's messages do nothing but clutter their mailboxes they can click on the "unsubscribe" link at the end of the message. After that click they are no longer customers of CNN.com.

CNN's has the potential to learn about customers even before they sign up for the service. Once they click on the banner to request the service, CNN can observe the article that they were reading before they click and use it to make inferences about the customer's interests. Learning continues with the initial questionnaires. CNN can keep track of which email is sent to each customer and refine their knowledge of the customers' preferences by observing their reactions to each message. Consider a customer who clicked on the "Full story" link in Figure 1-3. What email should CNN send them during the next interaction? Was the customer interested in Microsoft, the legal system, monopoly trials, or high-tech business news?

If CNN wants to maximize the probability that the customer will click return to their site during the next interaction they will send an email that is very similar to the previous one. It could be, for example, another piece of news about the Microsoft trial. Alternatively, they could send something about a different trial. If the customer is interested in the other trial, then CNN has learned that this person's interests were probably not related to Microsoft in particular, but to legal issues in general. However, it is also possible that the person was only interested in high-tech news and will not click on the email. Worse still, the new message could be perceived as junk mail and CNN might even lose a customer. How should a company such as CNN.com balance the objectives of providing good service in the present and learning so as to provide better service in the future? How should CNN simultaneously provide service while gathering information?

These are precisely the questions that are addressed in this dissertation. Chapter 2 provides a review of partially observable Markov decision processes (POMDPs) so as to provide the theoretical framework within which the analysis will take place. This chapter demonstrates the importance of POMDPs in Management Science and shows how the state-of-the-art solution techniques found in the Artificial Intelligence literature can be applied to managerial settings. This chapter also describes a new method to find approximate solutions to POMDPs. Chapter 3 directly addresses the problem of finding good service customization policies. Its results include a decision rule that allows for experimentation to learn about customers' tastes and quantifies the risk of losing the customer due to a bad service encounter. The model analyzed in this chapter allows for a quantification of the value of customers as a function of how well the company knows them. The value of the customer base is not adequately captured by the number of customers and the amount of time for which they have been with that company. Learning, therefore, is an important component of loyalty. Finally, chapter 4 analyses the costs and benefits of information acquisition. This chapter describes a decision rule that can be used by companies to determine whether they should use a service encounter to make a recommendation or to ask a question. The analysis also includes a model that determines how companies should balance investments in customer retention (through learning) and customer acquisition. The output of this

model gives insights into situations when information is most valuable, and describes how information acquisition policies change when the method of learning or the interface design policy is improved.

# Chapter 2

## Partially Observable Markov Decision Processes in Management Science

### 2.1 Introduction

Sequential decision making models have several applications in Management Science. These models are important to help managers make decisions in situations where present actions have an impact on future payoffs. Problems involving pricing decisions, inventory policies, and machine maintenance, for example, have been analyzed through these types of models. The objective in analyzing these systems is to obtain a policy, or control rule, that maps states of the world into concrete actions. These policies are usually found through dynamic programming. The Markov decision process, used to model problems of sequential decision making, is a well-known tool in the world of Operations Research and Management Science. It provides a rigorous analytical foundation for the analysis of repeated decisions where the decision-maker chooses between actions with different costs that have different impacts in the environment. The decision-maker then observes the effect on the environment and makes the next decision based on the new state.

There are many situations where the decision maker (DM) does not have a perfect view of the world. Managers rarely have complete information about their customers, their competition, or the production processes they use. In these situations, it is not sufficient to find the optimal actions given the state of the system because managers cannot determine the state of the system exactly. Markov Decision Processes are of limited use because they require the decision maker to have perfect knowledge of the environment. Partially Observable Markov Decision Processes (POMDPs) are a generalization of the MDP framework: the DM still chooses actions with different cost and impacts on the environment, but he doesn't know for sure what the environment is. The decision maker has a prior distribution over what the world should be, but he only observes it indirectly. There are two sources of uncertainty which are modeled explicitly: the underlying stochastic dynamical system and the DM's knowledge of that system's parameters.

Obtaining information is crucial in situations where parameters are not known perfectly.

The need for information is particularly acute when interactions involve people, and DMs must learn about customer behavior through interactions. The more DMs know the system (or the customer's preferences, as the case may be) the more likely they are to choose a good action (or provide good service). Actions have an impact on the system, and the DM observes a signal of that impact. After each observation, the DM updates estimates of some parameter and acts again. Parameter values about the system improve over time. The rate at which learning occurs depends on the actions taken. It would, therefore, be desirable to have a decision-making model that explicitly captures the value of learning.

The goal of maximizing learning often stands in conflict with the goal of maximizing immediate payoffs. Real learning only comes from surprises. The analytical framework must therefore balance the exploration/exploitation tradeoff. POMDPs are an appropriate framework. POMDPs have their origin in Electrical Engineering. More precisely, the application that motivated the development of the POMDP model was Drake's [29] problem of decoding information from a noisy channel. The next major contribution was Sondik's [86] analysis of the mathematical properties of the optimal solution and description of the first algorithm that finds exact solutions to POMDPs in a finite amount of time. Sondik's algorithm is extremely inefficient and infeasible for all but the smallest problems. Nevertheless, the theoretical importance of his work proved to be the foundation for almost all POMDP research that followed. Apart from Sondik's work, the 1970s and 1980s saw a great number of POMDP applications but little progress in solution methods (Monahan [70] and Cheng [23] being notable exceptions). Most applications made use of approximate solution methods the development of customized algorithms that exploit structural features of the specific problem being studied.

In the past decade or so, POMDPs have recently received a great deal of attention from the AI community because they provide a framework for the analysis of a number of robot navigation problems (e.g., [20], [58], [59]). This surge of interest resulted in the development of new and efficient solution techniques. These contributions significantly reduced the computational complexity involved in the computation of optimal policies for POMDPs. Exact solution methods are still impractical for most POMDP applications. However, the exact solution methods presented in the literature have provided the framework for a number of approximate solution methods. This chapter introduces one such approximation.

In addition to literature reviews contained in POMDP-related dissertations, there are a few published surveys about POMDPs. All these surveys have their merits and limitations. However, none of them fully accomplish the objective of explaining the ways in which POMDPs can help managers make decisions and understand the best available techniques for finding control policies. Some of them are outdated (e.g., [70], [62], [100]), and others focus exclusively on technical matters (e.g., [19]). Furthermore, the methodological gap between the solution methods used in Artificial Intelligence and Management Science has been considerably widened in recent years. The present chapter bridges the gap between the AI and the management literature with a focus on the managerial relevance of the present findings.



The remainder of this paper is organized as follows. §2.2 provides a formal definition of POMDPs making explicit the way in which they generalize MDPs. §2.3 provides the managerial motivation for studying POMDPs, by describing important managerial decision-making situations where fully observable MDPs would be inappropriate. The focus is on application in Management Science, but applications in other fields are also mentioned. §2.4 reviews the currently existing methods for solving POMDPs. Rather than providing and exhaustive listing of all the methods, the emphasis is on describing the algorithms that make important theoretical contributions to the understanding of the problem thus providing building blocks for the current state-of-the-art. §2.5 proposes a new algorithm for deriving control policies for POMDPs. Finally, §2.6 offers some concluding remarks and directions for further research.

## 2.2 Problem formulation

### 2.2.1 Markov Decision Processes

This section presents a description of Markov Decision Processes in order to review basic concepts and establish the notation and definitions that will be used elsewhere in this dissertation. The reader is referred textbooks such as [8] for a more thorough discussion on this subject.

**Definition 1** *A Markov Decision Process (MDP) is a 4-tuple  $\{\mathbf{S}, \mathbf{X}, T, R\}$  where*

- $\mathbf{S} = \{S_1, S_2, \dots, S_{|\mathbf{S}|}\}$  is a set of states; these are the possible states the system can be in at any point in time.  $S^0$  denotes the initial state and  $S^t$  denotes the state after  $t$  interactions.
- $\mathbf{X}(S), S \in \mathbf{S}$ , is a finite set of actions; these are the actions that the controller can take at each of the states.  $X^t$  denotes the action taken during the  $t$ 'th interaction.
- $T : \mathbf{S} \times \mathbf{X} \rightarrow [0, 1]^{|\mathbf{S}|}$  are the transition probabilities for each action in each state;  $T(S_1, X, S_2) = P(S_2|S_1, X)$  is the probability that the system will change to state  $S_2$  if action  $X$  is taken in state  $S_1$ .
- $R : \mathbf{S} \times \mathbf{X} \rightarrow \mathfrak{R}$  are the immediate rewards;  $R(S, X)$  is the reward obtained whenever action  $X$  is chosen when the system is in state  $S$ .

The Markov assumption, an essential property for the development of solution algorithms, states that the ability to act optimally does not depend on knowing what the previous states were or how the system arrived at state it is currently in. Mathematically, the Markov assumption is defined by the equation

$$P(S^{t+1} = S_i | S^t, X^t, S^{t-1}, X^{t-1}, \dots, S^1, X^1, S^0) = P(S^{t+1} = S_i | S^t, X^t).$$

It follows from the assumption that the DM only needs to consider the present state when choosing an action. All the relevant past information is summarized in the current state.

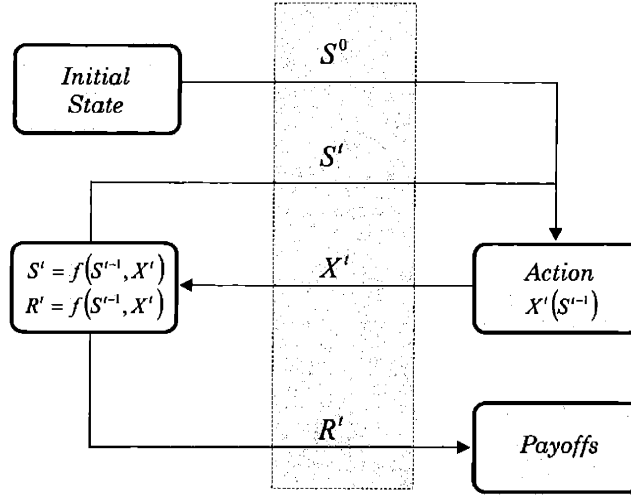


Figure 2-1: Sequence of events in a Markov decision process

The sequence of events in a MDP is depicted in Figure 2-1. At time  $t$ , the Decision Maker observes state  $S^t \in \mathbf{S}$  and takes action  $X^{t+1} (S^t) \in \mathbf{X}$  based on the state observed. He then received a feedback  $R^{t+1}$ , which is a function of  $X^{t+1}$  and  $S^t$ . The system then moves from  $S^t$  to  $S^{t+1}$  according to  $T(S^t, X^{t+1}, S^{t+1})$ . This cycle is repeated for  $N$  periods. In the case of infinite horizon problems ( $N = \infty$ ), a discount factor  $0 < \gamma < 1$  is usually introduced to ensure convergence. This discounted problem was first formulated by Blackwell [15]. The discount factor is not always necessary, but its absence requires the presence of additional structural features in the particular problem. One example of an infinite-horizon MDP that does not require a discount factor is the infinite-horizon version of the stochastic shortest path problem introduced by Eaton and Zadeh [31]. The discount factor is not necessary to ensure convergence of the finite horizon problem, but it often makes sense to have it for modeling assumptions (today's rewards are worth more than tomorrow's).

The control problem to be solved is the optimization of the total reward. The total reward is denoted  $V$  and is given by the expression

$$V = E \left\{ R_N (S^N) + \sum_{i=0}^{N-1} \gamma^i R [X^i, S^i] \right\}$$

where  $X_i$  is the action taken at time  $i$  and  $S_i$  is the state of the system at time  $i$ .

The DM's objective is to determine the actions that maximize the expected reward. Hence, the optimal value function is:

$$V^* = \max_{X^i \in \mathbf{X}} E \left\{ R_N(S^N) + \sum_{i=0}^{N-1} \gamma^i R[X^i, S^i] \right\}$$

**Definition 2** *A policy*

$$\mu : \mathbf{S} \rightarrow \mathbf{X}$$

maps each state  $S_i$  into an action  $X(S_i)$ .

The solution of an MDP is called the optimal policy. If the optimal policy is denoted  $\mu^*(S)$ , it follows that

$$V^* = \max_{X^i \in \mathbf{X}} E \left\{ R_N(S^N) + \sum_{i=0}^{N-1} \gamma^i R[\mu_i^*(S^i), S^i] \right\}$$

Traditional techniques for formulating and solving sequential optimization problems that make the Markov assumption is dynamic programming (e.g., [8]) Two basic solution techniques are policy iteration and value iteration. One important result from this literature is the principle of optimality (the formulation below is adapted from [8]):

**Theorem 3** *Principle of optimality.* Let  $\mu^* = \{\mu_1^*, \mu_2^*, \dots, \mu_{n-1}^*\}$  be an optimal policy for the basic problem, and assume that when using  $\mu^*$ , a given state  $S_k$  occurs at time  $k$  with positive probability. Consider the subproblem whereby we are at  $S_k$  at time  $k$  and want to minimize the “cost to go” from time  $k$  to the  $N$ :

$$E \left\{ R_N(S^N) + \sum_{i=k}^{N-1} \gamma^i R[\mu_i(S^i), S^i] \right\}$$

Then the truncated policy  $\{\mu_i^*, \mu_{i+1}^*, \dots, \mu_{n-1}^*\}$  is optimal for this subproblem.

There are two main solution methods for MDPs: policy iteration and value iteration. Both make use of the Bellman equation to develop iterative procedures that converge to the optimal solution. Which of the two methods is better depends on the structure of each particular problem. There have been several papers specializing these methods to particular problems. The most important solution methods for POMDPs are based on value iteration. A more precise description of this technique will be provided in §2.4.1 in the context of POMDPs.

## 2.2.2 Partially Observable Markov Decision Processes

POMDPs are a generalization of MDPs. In addition to the four elements of the Markov Decision Process defined in section 2.2.1,  $\{\mathbf{S}, \mathbf{X}, T, R\}$ , the POMDP also has: a discrete set of possible observations ( $\Theta$ ), and an observation model

$$W : \Theta \times \mathbf{S} \times \mathbf{X} \rightarrow [0, 1]$$

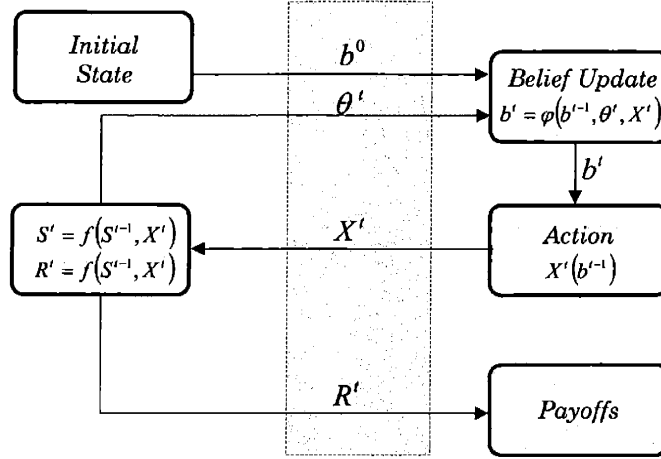


Figure 2-2: Sequence of events in a Partially Observable Markov Decision Process

defining the probability of making a particular observation given state/action pairs. Cassandra [19] provides the following concise and precise definition of POMDPs. His notation has been modified in order to be consistent with the terminology being used in this paper.

**Definition 4** A POMDP is a hextuple  $\{\mathbf{S}, \mathbf{X}, T, \mathbf{R}, \Theta, W\}$  where

$\mathbf{S}$  is a set of states;

$\mathbf{X}$  is a finite set of actions;

$T: \mathbf{S} \times \mathbf{X} \rightarrow [0, 1]^{|\mathbf{S}|}$  are the transition probabilities for each action in each state;

$R: \mathbf{S} \times \mathbf{X} \rightarrow \mathcal{R}$  are the immediate rewards;

$\Theta$  is a finite set of possible observations;

$W: \Theta \times \mathbf{S} \times \mathbf{X} \rightarrow [0, 1]$  is an observation model that defines the probability of making a given observation given a state-action pair.

The process  $\{\mathbf{S}, \mathbf{X}, T, R\}$  is often referred to as the “core process”. The sequence of events in a POMDP is depicted in Figure 2-2. The initial state of the system is  $S_0 \in \mathbf{S}$ . The decision maker cannot observe the state perfectly, but has prior beliefs about what the initial state may be. These beliefs are represented by a  $|\mathbf{S}|$ -dimensional vector  $b^0 \in \mathbf{B}$ , with each component corresponding to the probability that the customer belongs to a given segment. More specifically, if  $S^*$  denotes the true state of the system the prior beliefs are

$$b^0 = \begin{pmatrix} \Pr(S^* = S_1) \\ \Pr(S^* = S_2) \\ \dots \\ \Pr(S^* = S_{|\mathbf{S}|}) \end{pmatrix} := \begin{pmatrix} b_1^0 \\ b_2^0 \\ \dots \\ b_{|\mathbf{S}|}^0 \end{pmatrix}$$

where  $b_i^0 \in [0, 1] \forall i$  and  $\sum_{i=1}^{|\mathbf{S}|} b_i^0 = 1$ .

During each interaction, the DM chooses an action from the set  $\mathbf{X} = \{X_1, X_2, \dots, X_{|\mathbf{X}|}\}$ . Three events happen after this choice: the state of the system changes according to  $T(S, X)$ , the DM earns a revenue based on the function  $R(S, X)$  and the DM makes an observation  $\theta^t \in \Theta$ . After making the observation the DM updates the beliefs according to a function

$$\varphi : \mathbf{B} \times \mathbf{X} \times \Theta \rightarrow \mathbf{B}$$

that determines the new belief given the previous belief and the action/observation pair of the last interaction. If  $b^t$  denotes the belief vector after  $t$  interactions, then

$$b^t = \begin{pmatrix} \Pr(S^* = S_1 | b^0, X^1, \theta^1, X^2, \theta^2, \dots, X^t, \theta^t) \\ \Pr(S^* = S_2 | b^0, X^1, \theta^1, X^2, \theta^2, \dots, X^t, \theta^t) \\ \dots \\ \Pr(S^* = S_{|\mathbf{S}|} | b^0, X^1, \theta^1, X^2, \theta^2, \dots, X^t, \theta^t) \end{pmatrix} = \begin{pmatrix} b_1^t \\ b_2^t \\ \dots \\ b_{|\mathbf{S}|}^t \end{pmatrix}, \quad (2.1)$$

which can be written more compactly as

$$b^t = \begin{pmatrix} \Pr(S^* = S_1 | b^{t-1}, X^t, \theta^t) \\ \Pr(S^* = S_2 | b^{t-1}, X^t, \theta^t) \\ \dots \\ \Pr(S^* = S_{|\mathbf{S}|} | b^{t-1}, X^t, \theta^t) \end{pmatrix} = \begin{pmatrix} b_1^t \\ b_2^t \\ \dots \\ b_{|\mathbf{S}|}^t \end{pmatrix}$$

since  $b^{t-1}$  is a sufficient statistic for  $\{b^0, X^1, \theta^1, X^2, \theta^2, \dots, X^{t-1}, \theta^{t-1}\}$ , i.e.,

$$\Pr(S^* = S_i | b^0, X^1, \theta^1, X^2, \theta^2, \dots, X^t, \theta^t) = \Pr(S^* = S_i | b^{t-1}, X^t, \theta^t), \quad \forall i$$

The new belief can be computed in a Bayesian manner, using the relationship

$$\Pr(S^* = S_i | b^{t-1}, X^t, \theta^t) = \frac{1}{\Pr(\theta^t | b^{t-1}, X^t)} \cdot [b_i^{t-1} \cdot \Pr(\theta^t | S_i, X^t)]$$

where

$$\Pr(\theta^t | b^{t-1}, X^t) = \sum_{i=1}^{|\mathbf{S}|} [b_i^{t-1} \cdot \Pr(\theta^t | S_i, X^t)] \quad (2.2)$$

is a normalizing factor. The update function  $\varphi$  can then be defined by

$$\varphi(b, \theta, X) = \frac{1}{\Pr(\theta | b, X)} \cdot \begin{pmatrix} b_1 \cdot \Pr(\theta | S_1, X) \\ b_2 \cdot \Pr(\theta | S_2, X) \\ \dots \\ b_{|\mathbf{S}|} \cdot \Pr(\theta | S_{|\mathbf{S}|}, X) \end{pmatrix}. \quad (2.3)$$

It is useful to define a matrix

$$\Gamma^\theta(X) = \begin{bmatrix} \Pr(\theta|S_1, X) & 0 & \cdots & 0 \\ 0 & \Pr(\theta|S_2, X) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Pr(\theta|S_{|S|}, X) \end{bmatrix} \quad (2.4)$$

so that

$$\varphi(b, \theta, X) = \frac{b \cdot \Gamma^\theta(X)}{\Pr(\theta|b, X)}.$$

The DM's problem is to find a policy

$$\mu : \mathbf{B} \rightarrow \mathbf{X}$$

that determines the best action to take for each possible belief. The POMDP policy is much more complex than its MDP counterpart because the domain is no longer a finite set of states: it is a set of infinitely many beliefs. In the belief-state problem, the payoff function must be map belief states and actions to revenues, i.e.,

$$\tilde{R} : \mathbf{B} \times \mathbf{X} \rightarrow \mathfrak{R}$$

where  $\tilde{R}$  is defined by

$$\tilde{R}(b, X) = \sum_{i=1}^{|\mathbf{S}|} b_i \cdot R(S_i, X_j).$$

The optimal policy solves the equation

$$\mu^* = \arg \max_{X \in \mathbf{X}} \left\{ E \left( \sum_{t=0}^{\infty} \gamma^t r_t \right) \right\} \quad (2.5)$$

where  $\gamma \in (0, 1)$  is a discount factor and  $r_t$  is the reward at time  $t$ .

The policy that solves the POMDP  $\{\mathbf{S}, \mathbf{X}, T, \mathbf{R}, \Theta, W\}$  is the same policy that solves Markov Decision Process (MDP) is a 4-tuple  $\{\mathbf{B}, \mathbf{X}, \varphi, \tilde{R}\}$ . The state in the MDP  $\{\mathbf{B}, \mathbf{X}, \varphi, \tilde{R}\}$  is the continuous belief space, and the actions are the discrete set  $\mathbf{X}$ . Therefore, any solution method used for continuous-space discrete-action MDPs can be used to solve POMDPs. Indeed, early POMDP models (e.g., [29], [49]) were solved by discretizing the belief space and directly applying standard discrete MDP solution methods. Unfortunately, these methods are not very accurate efficient. Moreover, POMDPs have particular structural features that can be explored to develop better solution techniques. These features will discussed in §2.4.

## 2.3 Applications

### 2.3.1 Applications in Management Science

#### Machine Inspection and Quality control

Eckles [32] formulated a model of machine maintenance that proved to be an important building block for dozens of papers in the years that followed (e.g., [81], [80], [99]). Figure 2-3 depicts one version of the machine inspection problem. In this problem, the DM must decide whether to continue production or stop to inspect a machine. The state of the world is whether or not the machine works. The DM's payoffs depend on the yield (or quality of the output). After observing the machine's output, the DM updates the beliefs about the state of the machine and then makes the inspection decision for the following period. The objective is to maximize the yield over the planning horizon.

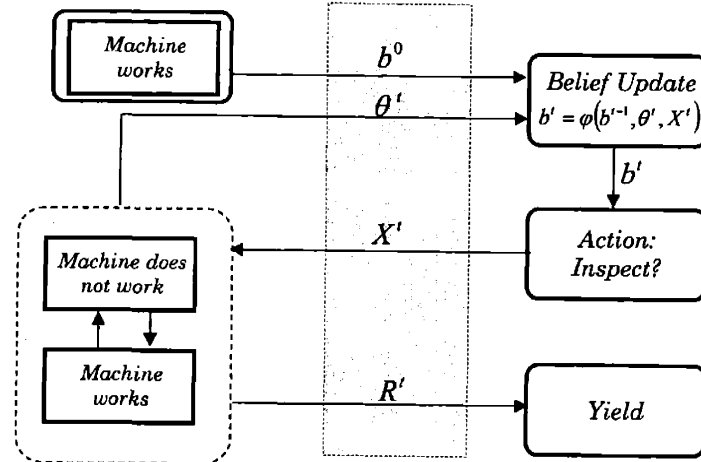


Figure 2-3: The machine inspection problem

The machine inspection problem has been extended to analyze quality control (e.g., [34]) and well as process control in general (e.g., [25]). This line of research continues to be popular, as evidenced by recent publications such as [25].

#### Demand Learning

Retailers operate in environments where they do not know the demand that they will have for each given product. The problem of determining optimal inventory levels under these conditions has been the subject of much investigation in operations management (e.g., [33]). These papers usually assume that the demand distribution is known, even though the exact demand is not. The resulting policies are the well-known  $(s, S)$  policies, which prescribe increasing the inventory up to level  $S$  if it is found to be lower than  $s$  at the beginning of

each planning period. This can be an unreasonable assumption in many situations. To remedy this shortcoming, Lovejoy presents a model where the demand distribution is unknown. The problem of determining optimal inventory levels can then be analyzed as a POMDP.

In Lovejoy's model, the observations are the inventory levels at the beginning of each planning period. The controls are the levels to which the inventory can be increased. Two observation models are proposed. In the first, firms observe sales but do not observe demand. In the second, firms observe demand. The difference between the two occurs when the inventory level reaches zero, in which case the firm that observes only sales does not receive any useful information after the stock runs out. The possible states of the world are the different possible demand probability distributions. After observing sales or demand, the firm updates the belief about the demand distribution. Lovejoy finds that the optimal control policy is a generalization of the  $(s, S)$  policy, where the levels  $s$  and  $S$  are functions of the current belief.

### **Service Customization**

POMDPs can be used to determine optimal service customization policies. This application is discussed in detail in the next chapter of this dissertation. In this application, the state space corresponds to different types of customers. The controls the firm has are the different ways in which the service can be customized. Customer types are defined by their preferences, i.e., by the way in which they react to each of the different services. Firms that provide services observe the way in which the customer reacts to the service and use this information to update the belief about the segment to which the customer belongs. The payoffs received by the firm depend on the utility the customer experienced from the customized service.

### **2.3.2 Applications in other fields**

Non-managerial applications are not directly relevant to this dissertation, but are mentioned here for two reasons. First, for the sake of completeness and in order to give the reader an idea of the vast applicability of POMDPs. Second, because engineering applications have motivated important innovative solution approaches that represent major contributions to the theory of POMDPs. Drake's [29] application is historically significant because it motivated the development of the first POMDP model and is described in detail in §2.3.2. § 2.3.2 provides a brief overview of other applications.

#### **Decoding signals**

Drake [29] studied the problem of controlling a symmetric two-state Markov source through a memoryless binary symmetric channel. The system studied by Drake is depicted in Figure 2-4 (note that Drake's original notation has been altered in order to preserve the consistency



of notation of the present paper). Time progresses in discrete units, and the state space is  $S = \{S_1, S_2\}$ . The transition functions are symmetric and defined as follows:

$$\begin{aligned} \Pr(S_{t+1} = S_1 | S_t = S_1) &= \beta & \Pr(S_{t+1} = S_1 | S_t = S_2) &= 1 - \beta \\ \Pr(S_{t+1} = S_2 | S_t = S_2) &= \beta & \Pr(S_{t+1} = S_2 | S_t = S_1) &= 1 - \beta \end{aligned}$$

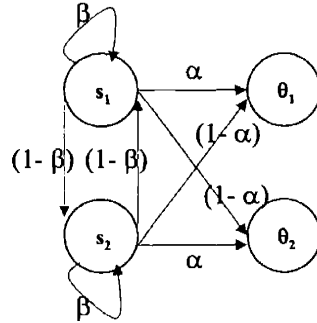


Figure 2-4: Drake's symmetric two-state Markov source monitored by a binary symmetric channel

There are three possible actions at any of the two states.  $\mathbf{X} = \{X_1, X_2, X_3\}$ .  $X_1$  consists of asserting that the current state is  $S_1$ ;  $X_2$  consists of asserting that the current state is  $S_2$ ;  $X_3$  consists of ascertaining the present state. The cost structure associated with these actions is also symmetric, and it is given by the three equations below (where  $L$ ,  $K$ , and  $B$  are positive constants).

$$\begin{aligned} R(S_1, X_1) &= L & R(S_2, X_1) &= -K \\ R(S_1, X_2) &= -K & R(S_2, X_2) &= L \\ R(S_1, X_3) &= -B & R(S_2, X_3) &= -B \end{aligned}$$

If the problem was completely observable, the optimal solution would have been trivially simple: choose  $X_1$  if the state is  $S_1$  and  $X_2$  if the state is  $S_2$ . The complication, however, is that the states are not directly observable. The states  $S_1$  and  $S_2$  cannot be observed directly. The observer either receives a signal from the set  $\Theta = \{\theta_1, \theta_2, \theta_3\}$  with the following probabilities:

$$\begin{aligned} \Pr(\theta_1 | S_1, X_1) &= \Pr(\theta_1 | S_1, X_2) = \alpha & \Pr(\theta_2 | S_2, X_1) &= \Pr(\theta_2 | S_2, X_2) = \alpha \\ \Pr(\theta_2 | S_1, X_1) &= \Pr(\theta_2 | S_1, X_2) = (1 - \alpha) & \Pr(\theta_1 | S_2, X_1) &= \Pr(\theta_1 | S_2, X_2) = (1 - \alpha) \\ \Pr(\theta_3 | S_1, X_i) &= 1 \text{ if } i = 3, 0 \text{ otherwise} & \Pr(\theta_3 | S_1, X_i) &= 0 \quad \forall i \end{aligned}$$

The objective function is given below:

$$\max_{X^i \in \mathbf{X}} E \left\{ \sum_{i=0}^N \gamma^i R[X^i, S^i] \right\}$$

Drake analyzes two separate cases:  $\alpha = \frac{1}{2}$  and  $\alpha > \frac{1}{2}$ . The first case consists of essentially ignoring the observations. In this case, the optimal policy consists of periodically choosing  $a_3$  to determine the current state, the length of these periods being a function of the transition probability  $\beta$ . The second case is much more difficult. There is no closed form solution, but the analysis of the problem as a dynamic program and numerical solutions of a few special cases reveal the important insight that the controller’s state of knowledge plays a key role in determining the action to be taken in each point in time. Drake notes that it is “the steady state statistics of these statistical state variables which allow one to determine the properties of a Bayesian decoder which operates on the infinite past.” However, it was Astrom [6] who presented a thorough understanding and formalization of these statistics and their role in the solution of POMDPs by redefining the POMDP as a fully observable “Belief State MDP” (c.f. §2.2.2).

## Other Applications

One of the first applications of POMDPs was in the context of education. Smallwood and Sondik [85] studied a problem where the state of the world was whether or not the student possessed a certain knowledge. Knowledge could only be observed indirectly, through tests whose implementation had costs. The controls are the various ways in which the material could be presented to the student.

Lane [53] describes a POMDP application in the British Columbia’s salmon fishing industry. The DM is a fisherman that must determine in which zone to fish. The objective is to maximize the net operating income, which takes into account the cost of fishing and the price of fish. The decision is made under imperfect information because the fisherman does not know the catch potential in a given region, which is a function of the number of fish in the region and “catchability”. The core process is a Markov process that determines the amount of fish in each potential region. The observations are the total catch (by weight) of a given expedition in a particular region. These observations are used to update the priors about the amount of fish in each region. The optimization problem was formulated as a finite horizon POMDP, where the number of time periods is the number of fishing expeditions left in the season.

POMDPs can be used to decide where to search for a moving target. Pollock [73] formulated the original problem where the target could be in only one of two places, and Eagle [30] generalized it to the case where the target could be in any of a finite number of cells and the set of cells available for searching depend on the cell that was searched in the previous period. The objective is to find the sequence of search cells that maximizes the chance of finding the target in a fixed number of periods.

Hauskrecht [43] demonstrated how POMDPs can be used as a decision-support tool in the treatment of ischemic heart disease (IHD). The state of the system is the patient’s condition. The patient can be dead, alive with IHD and alive without IHD. The controls are the different types of treatment, which can include do nothing, administer medication,

or request an angiogram. Doing nothing gives the physician the opportunity to observe symptoms in an unobtrusive manner, but increases the chance of a myocardial infarction (MI) or death. An angiogram provides a great amount of information, but in addition to causing patient discomfort it may also lead to MI or death. Finally, administering medication can ease the symptoms, obscuring information about the true cause of the patient's initial discomfort and can also have negative side effects. The physician's decision must be made in light of past observations of symptoms as well as the ways in which the disease can progress. The optimal control is the one that maximizes objective functions such as increasing length and /or quality of life, or decreasing overall discomfort.

The control of robots in partially observable environments has been a very popular research topic in Artificial Intelligence in recent years (see [48] for an overview). These problems are motivated by the wide application potential of autonomous robots that are often unable to determine their location due to mechanical and sensorial limitations. The objective is for the robot to accomplish a given task (e.g., to reach a given location) as fast as possible (or to maximize the probability that the location is reached within a given number of periods). The observations are the signals obtained by the robot's sensors, and the controls are the directions in which the robot can move.

## 2.4 Solution Methods

This section places an emphasis on exact solution methods for three reasons. First, because these methods represent the most significant theoretical contributions and led to a better conceptual understanding of the problems. Second, because approximate solution methods are numerous and often involve "ad-hoc" techniques. Approximate solution methods come from a wide variety of fields and involved many different techniques, and their study does not necessarily lead to a more precise understanding of the main issues at stake in the solution of POMDPs. Third, these methods provide a logical sequence to introduce the approximation method of § 2.5.

### 2.4.1 Value Iteration

Solution methods for POMDPs are based on value iteration. Therefore, this section briefly reviews the basic ideas behind this solution technique.

The optimal policy for POMDPs from (2.5) must satisfy the Bellman equation

$$V^*(b) = \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \sum_{b' \in \mathbf{B}} P(b'|b, X) V^*(b') \right\},$$

i.e.,

$$\mu^*(b) = \arg \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \sum_{\theta \in \Theta} P(\theta|b, X) V^*(\varphi(b, \theta, X)) \right\}.$$

When there is a finite amount of possible observations, the Bellman equation can also be written as

$$V^*(b) = \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \sum_{\theta \in \Theta} P(\theta|b, X) V^*(\varphi(b, \theta, X)) \right\}.$$

The mapping

$$(TV)(b) = \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \sum_{\theta \in \Theta} P(\theta|b, X) V(\varphi(b, \theta, X)) \right\} \quad (2.6)$$

is called the dynamic programming mapping. Note that  $(TV)(\cdot)$  is itself a value function defined over  $\mathbf{B}$ , so  $T$  is a mapping that transforms the value function  $V$  into another value function  $TV$ . Using this notation, (2.6) can be rewritten as

$$TV^*(b) = V^*(b).$$

The dynamic programming mapping has two important properties: it is monotonic and it is a contraction under the supremum metric.

**Definition 5** A mapping is a *contraction under the supremum metric* if  $\exists c \in [0, 1)$  such that

$$|TU(b; X) - TV(b; X)| \leq c \cdot \sup_{b \in \mathbf{B}} |U(b; X) - V(b; X)|$$

for all  $U, V \in \mathcal{V}$ , for all  $b \in \mathbf{B}$  and all  $X \in \mathbf{X}$ , where  $\mathcal{V}$  is the set of all bounded functions from  $\mathbf{B}$  to  $\mathfrak{R}$ .

**Definition 6** A function  $T$  is *monotonic* if

$$U(b) \geq V(b) \rightarrow TU(b) \geq TV(b)$$

holds for all  $U, V \in \mathcal{V}$

The fact that  $T$  is a contraction mapping means that it satisfies the fixed point theorem, stated below.

**Theorem 7 (Fixed point theorem)** If  $\mathcal{V}$  is a complete metric space and  $T$  is a contraction of  $\mathcal{V}$  into  $\mathcal{V}$  then there exists a unique  $V \in \mathcal{V}$  such that  $T$

$$TV(b) = V(b), \forall b \in \mathbf{B}$$

**Proof.** This is a standard result in the theory of contraction mappings. Many textbooks such as Rudin [83] present a complete proof of this theorem. ■

Denardo [26] proved one important consequence of Theorem 7: the repeated application of the operator  $T$  to a value function  $V$  converges to the optimal value function. This result is stated as a theorem below.

**Theorem 8** *If  $T^n$  denotes  $n$  repeated applications of the operator  $T$  then*

$$\lim_{n \rightarrow \infty} (T^n V)(b) = V^*(b)$$

where  $V^*(b)$  is the optimal value function.

**Proof.** See [26]. ■

Theorem 8 provides the basis for the value iteration algorithm, which consists of evaluating a sequence of value functions  $V^n = TV^{n-1}$  until

$$\sup_{b \in \mathbf{B}} |V^*(b) - V^n(b)| < \varepsilon$$

where  $\varepsilon$  is an arbitrarily small error factor known as the Bellman error. In order to solve the infinite horizon problem, the operator  $T$  is applied to the value function  $V^1$  until  $|V^*(b) - V^n(b)| < \varepsilon$ , which can be expressed as a function of  $|V^n(b) - V^{n-1}(b)|$  (this is a standard result, found in [9] or [58], for example).

## 2.4.2 General Properties of POMDPs

### Properties of the Optimal Value Function

The most important mathematical property of POMDPs is that their finite-horizon value function is piecewise linear and convex. This implies that the optimal value function for the infinite horizon problem can be approximated arbitrarily well by a piecewise linear convex function. To be exact, the value function after  $n$  iterations of the value iteration algorithm can be written as

$$V^n(b) = \max_{\alpha \in A^n} (b \cdot \alpha)$$

where  $A^n$  is a finite set of  $|\mathbf{S}| - \text{dimensional}$  vectors.

The piecewise linearity and convexity of the value function is stated in the form of a theorem below.

**Theorem 9** *The finite horizon value function is piecewise linear and convex for every horizon length.*

**Proof.** The first complete proof can be found in [86]. Two simpler proofs are briefly described below.

**Proof 1:** Show that the value function for  $t = 1$  is piecewise linear and convex. Prove by induction that if the value function for  $t = i$  is piecewise linear and convex then the value

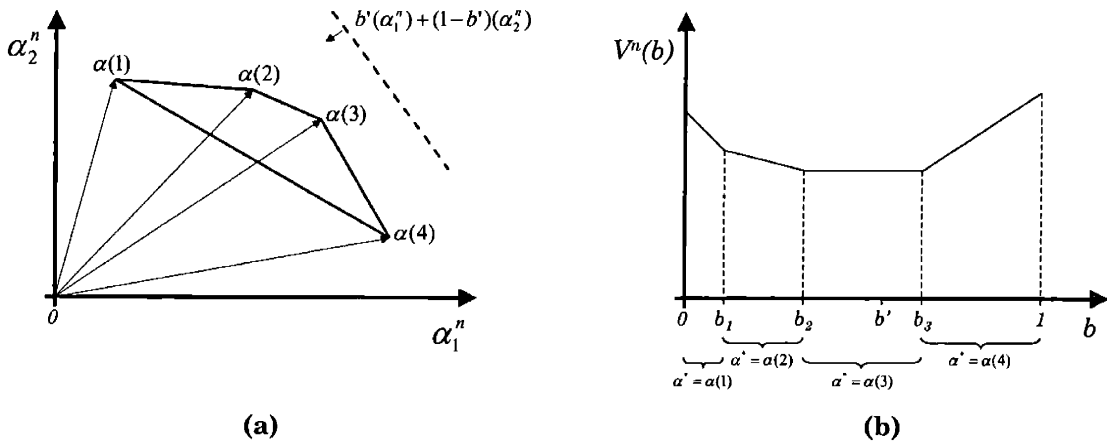


Figure 2-5: A set  $A^n$  of vectors and their corresponding value function

function for time  $t = i + 1$  is also piecewise linear and convex. (See [19]) An important part of the proof is showing that  $V^{i+1}(b)$ , the value function for step  $(i + 1)$ , can be expressed in terms of  $A^i$ , the vectors defining the value function  $V^i$ . The precise relationship is the following (c.f. [44]):

$$V^{i+1}(b) = \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \sum_{\theta \in \Theta} \max_{\alpha_i \in A^i} \sum_{S' \in \mathbf{S}} \left[ \sum_{S' \in \mathbf{S}} \Pr(S', \theta | S, X) \right] \alpha_i(S') \right\}$$

Proof 2: There is a finite number of policy trees, each region corresponds to a different policy tree. (See [58]) ■

Figure 2-5a shows an example of the convex hull of a hypothetical set  $A^n = \{\alpha(1), \alpha(2), \alpha(3), \alpha(4)\}$  and 2-5b shows the corresponding value function  $V^n(b)$ . The belief states labeled  $b_1, b_2,$  and  $b_3$  are the boundary points of the four linear segments that make up the value function. The result of theorem 9 implies that the value function has a compact representation as a set of vectors, and in Figure 2-5 there are four such vectors, each corresponding to one of the four linear segments. This is a very important result that plays a key role in the development of the algorithms that follow: the value function corresponding to a POMDP is a continuous function in  $[0, 1]^n$  (where  $n$  is the number of states in the underlying dynamical system), yet it can always be represented by a finite set of vectors.

Let  $A^t$  denote the set of the vectors that are necessary to describe the  $t$ -step value function.  $A^t$  is sometimes called the support set of the value function. In Figure 2-5,  $A = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ . The value function can be completely described by

$$V^*(b) = \begin{cases} b \cdot \alpha_1 & \text{if } 0 \leq b < b_1 \\ b \cdot \alpha_2 & \text{if } b_1 \leq b < b_2 \\ b \cdot \alpha_3 & \text{if } b_2 \leq b < b_3 \\ b \cdot \alpha_4 & \text{if } b_3 \leq b < 1 \end{cases}$$

More generally, the optimal value function can be written as

$$V^*(b) = \max_{\alpha_i \in A^*} b \cdot \alpha_i$$

A subset  $S(\hat{\alpha}, A^t)$  of  $[0, 1]^{|S|}$  is called the support region for  $\hat{\alpha} \in A$  if

$$S(\hat{\alpha}, A^t) = \left\{ b \in [0, 1]^{|S|} : b \cdot \hat{\alpha} \geq b \cdot \tilde{\alpha} \forall \tilde{\alpha} \in A^t \right\}.$$

### Why are POMDPs difficult?

Theorem 9 provides the missing link to ensure that the value iteration algorithm is certain to arrive at the optimal solution for the finite-horizon problem and an arbitrarily good approximation for the infinite horizon problem in a finite amount of time. Unfortunately, this finite amount of time can be very large and often impractical. The essential difficulty of solving POMDPs lies in the efficient computation of the sets  $A^n$  since the size of these sets grows very fast with each step of the value iteration algorithm. More precisely, in each iteration of the algorithm there is one new  $\alpha$  vector for each possible product recommendation  $X$  for each permutation of size 2 of the vectors in  $A^{n-1}$ .

Each  $\alpha$  vector corresponds to a policy tree and neighbouring  $\alpha$ 's correspond to different policy trees. Let  $\delta_i = \delta(\alpha_i)$  denote the policy tree corresponding to  $\alpha_i$ . Each policy tree corresponds to a unique vector, so the mapping  $\delta_i \rightarrow \alpha_i$  is one-to-one and the inverse mapping  $\alpha_i = \alpha(\delta_i)$  is well-defined. The vectors  $\alpha_i$  have  $|S|$  dimensions:  $\alpha_i = (\alpha_i^1, \alpha_i^2, \dots, \alpha_i^{|S|})$ . Each component  $\alpha_i^s$  represents the value of implementing  $\delta_i = \delta(\alpha_i)$ , the policy tree corresponding to  $\alpha_i$ , when the true initial state of the system is  $s$ . It will sometimes be more convenient to use the notation  $\alpha^s(\delta_i)$  instead of  $\alpha_i^s$ , but both mean the same thing.

Let  $\delta'_i$  denote the first action prescribed (the action in the parent node) by policy tree  $\delta_i$ . If  $\delta_i$  is a  $t$ -step policy tree then let  $\delta_i(\theta_j)$  be the  $(t-1)$ -step policy tree to be followed if  $\theta_j$  is observed after taking action  $\delta'_i$ .

We are now in a position to state precisely how a  $t$ -step vector  $\alpha \in A^t$  can be defined in terms of vectors in  $A^{t-1}$  and some parameters.

$$\alpha_i^s = R(\hat{s}, \delta'_i) + \sum_{j=1}^{|\Theta|} \left( \sum_{k=1}^{|S|} \alpha[\delta_i(\theta_k)] \cdot T[\hat{s}, \delta'_i, s_i] \cdot W[s_i, \delta'_i, \theta_j] \right)$$

where  $T$  and  $W$  are as defined in Definition 4.

Solving a POMDP reduces to finding the optimal set  $A^*$ . Each  $\alpha$  in the set  $A$  corresponds to a different policy. Expressing the value function in terms of a finite set of vectors paves

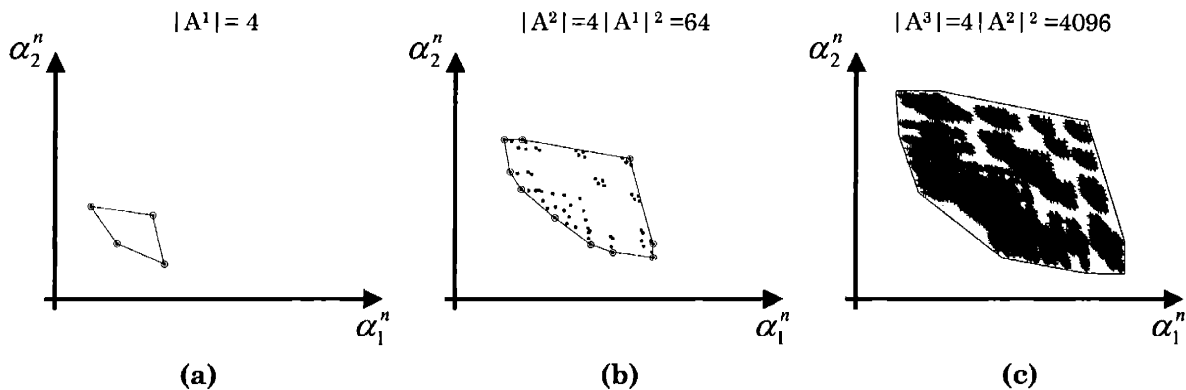


Figure 2-6: Generating sets  $A^n$  from  $A^{n-1}$

the way for a number of variations on the value iteration method. POMDPs can be solved by finding  $A_0^*$ , the set of the optimal vectors  $\alpha$  for the 1-step problem. This set can then be used to build  $A_1^*$ , and so on.

The way in which the sets  $A^n$  generated make it impossible to determine the optimal value function in closed-form, and the rate at which they grow often make numerical solutions impractical. Figure 2-6 show the convex hull of a sequence of sets  $A^n$  as  $n$  increases. The next section describes solution approaches that can be used in absence of a closed-form solution. These methods basically consist of making the generation of the sets  $A^n$  more efficient in order to find the optimal solution numerically.

### 2.4.3 Exact Solution Methods

This section describes five different algorithms that can be used to find exact solutions to POMDPs. Figure 2-7 shows how they are related. The first algorithm, by Sondik [86], is very inefficient but contains the foundation upon which all other algorithms are built. Monahan [70] improved Sondik's algorithm by specifying a linear program to remove extraneous vectors between value iteration updates. Eagle [30] made a small improvement to Monahan's algorithm that led to important insights on the structure of POMDPs. Cheng's [23] linear support algorithm begins by specifying an approximate value function over the entire belief space and proceeds by improving the value function at each step of the algorithm. One important property of this algorithm is that it can be stopped before optimality is reached in order to obtain good suboptimal policies. Finally, Littman's [58] Witness Algorithm draws on recent work on artificial intelligence to improve on Cheng's value function improvement procedure.



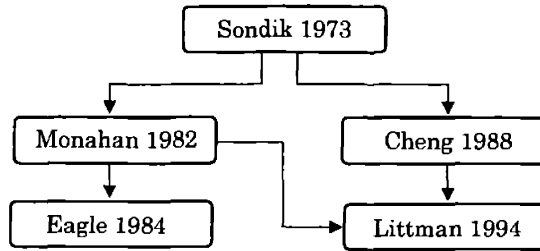


Figure 2-7: Algorithms for POMDPs.

### Sondik’s Algorithm

Sondik’s One-pass algorithm was regarded as ground-breaking work because it was the first exact algorithmic solution to POMDPs. The main sequence of steps defining the algorithm are described below.

**Algorithm 10** (by Sondik [86] ; as described in [19])

- (1) Initialize a search list of belief states to contain any single point on the belief simplex. Initialize a set of vectors,  $\hat{\mathcal{V}}_t$ , to be empty.
- (2) Remove a point from the search list. If the list is empty, then we are finished and  $\hat{\mathcal{V}}_t = \mathcal{V}_t^*$ .
- (3) Find the true vector (and its associated action) for this point and add it to  $\hat{\mathcal{V}}_t$ .
- (4) Define a region around this point where this vector is guaranteed to be the true vector.
- (5) Select points that lie on the edges of this region and add them to the search list.
- (6) Go to step (2).

The next two algorithms take different approaches to the problem of generating the set  $A_{t+1}$  from the set  $A_t$ . First, Monahan creates a set  $\tilde{A}_{t+1} \supseteq A_{t+1}$  and eliminates the unnecessary vectors from  $\tilde{A}_{t+1}$ , eventually reducing it to  $A_{t+1}$ . Second, Cheng generates a sequence of sets  $A_{t+1}^1 \subseteq A_{t+1}^2 \dots = A_{t+1}$  that converge in a finite amount of time to the optimal set  $A_{t+1}$ .

### Monahan’s Algorithm

Monahan’s [70] algorithm is quite simple to understand conceptually. He uses an enumeration technique to generate a set containing every possible policy tree in a given state. Then, he uses a linear programming approach to reduce this set to the trees corresponding the  $\alpha$  vectors contained in the optimal set  $A^*$ . The most important result from Monahan is the formulation of a linear program (describe more clearly in [58] or [19]) that can determine whether or not a policy is extraneous. Interestingly, Monahan attributes this LP to Sondik, but Sondik’s algorithm did not contain this important step.

Extraneous policies are those that cannot be part of the optimal solution. Policies can be extraneous for one of three reasons (see Littman [58] p.20 for details). (1) It is strictly dominated by another policy for every belief; (2) It is optimal only over a set of measure zero, (3) It generates exactly the same value function as another policy tree (i.e., a tie, the policy tree is redundant). The linear program used to find whether the vector  $\hat{\alpha}$  coefficients is extraneous in the set  $A_n$  is the following:

$$\begin{aligned}
& \min_{b \in B} x - b \cdot \hat{\alpha} \\
& \text{s.t.} \\
& x \geq b \cdot \alpha', \text{ for all } \alpha' \in A_n, \alpha' \neq \hat{\alpha} \\
& \sum_{i=1}^{|S|} b_i = 1 \\
& b_i \geq 0, \forall i
\end{aligned}$$

where  $B$  is the belief space. If  $x \neq 0$ , then remove  $\hat{\alpha}$  from  $A_n$ . If  $x - b \cdot \hat{\alpha} \geq 0$ , it means that  $\hat{\alpha}$  is dominated for every possible belief state.

### Eagle's algorithm

Eagle [30] algorithm is not significantly different from Monahan's algorithm and should perhaps be referred to as "Eagle's variant of Monahan's algorithm". This algorithm is mentioned here because the small modifications can lead to insights that help understand POMDPs. In particular, Eagle formulated the dual of Monahan's linear program, which has a rich geometric interpretation which shed some light into what the linear program actually does. The dual formulation is the following:

$$\begin{aligned}
& \max \nu & (2.7) \\
& \text{s.t.} \\
& \sum_{i=1}^{|A_n|-1} \lambda_i \alpha_i - \nu \geq \hat{\alpha} \\
& \sum_{i=1}^{|A_n|-1} \lambda_i = 1 \\
& \lambda_i \geq 0, \forall i
\end{aligned}$$

The subscripts  $i = 1, 2, \dots, (|A_n| - 1)$  index all the  $\alpha$  vectors in  $A_n$  except the vector being tested,  $\hat{\alpha}$ . If  $\nu \geq 0$ , then  $\hat{\alpha}$  is dominated and should be removed from  $A_n$ . If linear program is feasible, the first series of inequality constraints imply that there exists a linear

combination  $\left( \sum_{i=1}^{|A_n|-1} \lambda_i \alpha_i \right)$  of the elements in  $A_n$  excluding  $\hat{\alpha}$  such that  $\sum_{i=1}^{|A_n|-1} \lambda_i \alpha_i \geq \hat{\alpha}$ . If such a combination exists, then  $\alpha$  is inside the convex hull of the set  $A_n \setminus \hat{\alpha}$  ( $A_n$  without  $\hat{\alpha}$ ). Figure 2-8 illustrates two cases where the set  $A_n \setminus \hat{\alpha} = \{\alpha(1), \alpha(2), \alpha(3)\}$ . In the first one, the vector being tested is  $\hat{\alpha} = \alpha'$  and it is not needed in the optimal solution. In the second, the vector being tested is  $\hat{\alpha} = \alpha''$  and it is potentially useful in the optimal solution.

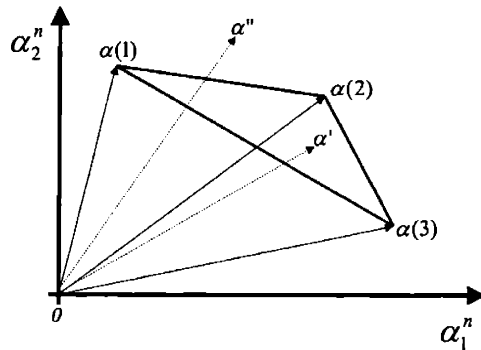


Figure 2-8: Geometric interpretation of Eagle's linear program

### Linear Support Algorithm

Cheng's [23] linear support algorithm generates at each step  $t$  a sequence of sets of  $\alpha$  vectors that converges in a finite amount of time to the optimal set. An example makes it easy to understand what is going on. Let  $A_t^0$  denote the set of  $\alpha$ 's for the endpoint of the belief space. In a two dimension belief state, a plot of the value function induced by  $A_t^0$  looks like the one in figure 2-9.

Cheng's algorithm finds the belief states (points) that separate the regions and generates the optimal  $\alpha$  vectors for those points. In figure 2-9, there is only one such point, which is marked  $b'$ . Call its corresponding optimal vector  $\alpha'$ . The algorithm then updates  $A_t^0$  to  $A_t^0 \cup \alpha' := A_t^1$ . The value function induced by  $A_t^1$  could look like the one in figure 2-10. The procedure described above for  $b'$  would then be repeated for boundary points  $b_1''$  and  $b_2''$ . The process is repeated until value function can't be improved, at which time we are certain to have  $A_t^*$ .

$A_t^*$  is then used to calculate the optimal  $\alpha$  at the boundary points for  $A_{t+1}^0$  and so on...

The problem with this algorithm is that it takes too long to check every boundary point. It doesn't look that way in the simple 2-dimensional illustrations in the section, but iterations can take a long time in higher dimensions. The witness algorithm provides a way of identifying belief points where the current set of vectors is not optimal, thus eliminating the need to check every boundary point.

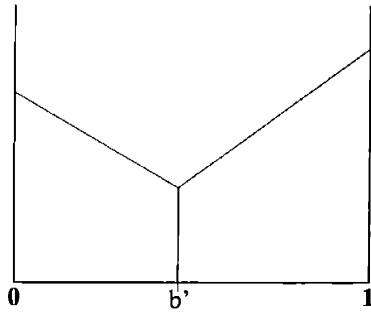


Figure 2-9: A plot of the value function  $V_t^0 = \max_{\alpha \in A_t^0} b \cdot \alpha$

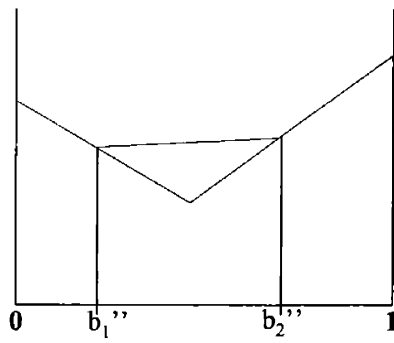


Figure 2-10: A plot of the value function  $V_t^1 = \max_{\alpha \in A_t^1} b \cdot \alpha$

## The Witness Algorithm

Littman's [58] Witness algorithm is one of the most important recent contributions to the efficient generation of  $A^n$  from  $A^{n-1}$ . The Witness algorithm efficiently builds a set  $Q^n$  such that  $Q^n \supseteq A^n$  and then uses Monahan's linear program (2.7) or an equivalent procedure to reduce  $Q^n$  to  $A^n$ . The set  $Q^n$  is defined as

$$Q^n = \bigcup_{X \in \mathbf{X}} Q^n(X)$$

where each set  $Q^n(X)$  contains the vectors  $\alpha$  corresponding to the q-function defined by product  $X$ . More precisely,

$$q^n(b, X) = \max_{\alpha \in Q^n(X)} (b \cdot \alpha),$$

and the q-function  $q^n(b, X)$  is the payoff obtained when the agent recommends product  $X$  for the current period and acts optimally thereafter. Two properties immediately follow. First,

$$\max_{X \in \mathbf{X}} [q^n(b, X)] = V^n(X), \quad (2.8)$$

implying the necessary conditions that  $Q^n \supseteq A^n$ . Second, the elements of  $Q^n(X)$  can be expressed as

$$\alpha_k(X) = p^A(X) + \gamma \sum_{\theta \in \Theta} \Gamma^\theta(X) \alpha_k^\theta(X)$$

where the matrices  $\Gamma^\theta(X)$  are defined as in (2.4) and  $\alpha_k^\theta(X) \in A^{n-1}$  for all  $k$  and for all  $\theta$ .

The sets  $Q^n(X)$  are built by iteratively adding new elements to a sequence of sets  $\hat{Q}^n(X)$ . The Witness theorem establishes a condition for determining whether or not  $\hat{Q}^n(X) \supseteq Q^n(X)$  and the iterative process can be stopped.

**Theorem 11** (*Witness Theorem*) *The true q-function*

$$q^n(b, X) = \max_{\alpha \in Q^n(X)} (b \cdot \alpha)$$

*differs from the approximate q-function*

$$\hat{q}^n(b, X) = \max_{\alpha \in \hat{Q}^n(X)} (b \cdot \alpha)$$

*if and only if there exists some  $\alpha_k(X) \in \hat{Q}^n(X)$ ,  $\theta \in \Theta$ ,  $\alpha^i \in A^{n-1}$  for which there exists a belief  $b$  where the condition*

$$b \cdot [\Gamma^\theta(X) \alpha_k^\theta(X)] < b \cdot [\Gamma^\theta(X) \alpha^i]$$

*holds.*

**Proof.** See Littman [58]. ■

The optimality conditions of the Witness Theorem can be verified by means of a linear program.

$$\begin{aligned}
& \max (b \cdot \beta) \\
& \text{s.t.} \\
& \quad b \cdot \alpha_k(X) \geq b \cdot \hat{\alpha}_k(X), \forall \hat{\alpha}_k(X) \in \hat{Q}^n(X) \\
& \quad \sum_{\forall i} b_i = 1 \\
& \quad b_i \geq 0, \forall i
\end{aligned}$$

where

$$\beta = [\Gamma^\theta(X) \alpha^i - \Gamma^\theta(X) \alpha_k^\theta(X)]$$

There will be one linear program for each  $\alpha_k(X) \in \hat{Q}^n(X)$ , for each  $\theta \in \Theta$ , and for each  $\alpha^i \in A^{n-1}$ . If the objective function is positive, then the corresponding belief  $b$  is a “witness” (hence the name of the theorem) to the fact that  $\hat{Q}^n(X)$  is not optimal. A new vector  $\alpha^{new}(X)$  is then added to  $\hat{Q}^n(X)$ , where the components of  $\alpha^{new}(X)$  are the same as those of  $\alpha_k(X)$  except for  $\alpha_k^\theta(X) = \alpha^i$ . The process is repeated until no more witness points can be found and it can therefore be asserted that  $\hat{q}^n(b, X) = q^n(b, X)$ .

The Witness theorem establishes the important complexity result that any necessary improvements to the q-functions can be made by solving at most  $|\hat{Q}^n(X)| \cdot |\Theta| \cdot |A^{n-1}|$  linear programs. The equivalent problem of improving the value function directly is NP-complete. Even though the number of computations necessary to calculate the optimal q-functions can be quite large, the running time of the algorithm can be bounded by a polynomial in the size of  $|\hat{Q}^n(X)|$ , and empirical tests show that the algorithm performs very well in practice, in particular when  $|\Theta|$  is small.

### Other exact solution methods

There are a few other exact algorithms in the literature. These will not be discussed in detail because they are essentially variants of the algorithms described above. The reader is referred to the original references for the details of these algorithms. First, there is Sawaki’s [84] partition algorithm. This algorithm is of theoretical interest in dynamic programming because it is based on techniques that can be applied to any piecewise linear value function (see [17] or [23] for details). For our purposes, however, it is essentially the same as Sondik’s algorithm. The only difference is the way in which the partitions and the support sets are constructed. Second, there is Cheng’s [23] relaxed region algorithm. This algorithm is also a minor variation of Sondik’s algorithm. In this case, the difference is that support regions are defined in a more efficient manner. Finally, there is incremental pruning ([21]; [103] improves upon the initial formulation). In this algorithm, the stages of generating and testing  $a$  vectors are interleaved. Their algorithm is designed so that it is sometimes possible to avoid generating vectors that are certain to be useless by “pruning” sets of partially constructed vectors.

## 2.4.4 Approximate Solution Methods

There are several approximate solution methods proposed in the literature. Many of them consist of applying an exact solution method and stopping the algorithm before it is finished. These methods are based on Bellman error method. The Bellman residual theorem provides the theoretical foundation for algorithms relying on the Bellman error method. The formulation below is from [58].

**Theorem 12 (Bellman residual):** *If the maximum difference between  $V_{t-1}$  and  $V_t$  (sometimes called the Bellman residual of  $V_{t-1}$ ) is less than  $\varepsilon$ , then the reward gathered by the greedy policy on either  $V_{t-1}$  and  $V_t$  never differs from that of the optimal policy by more than  $\frac{2\varepsilon\gamma}{(1-\gamma)}$  at any belief state.*

Another class of approximations consist acting greedily with respect to approximate value functions. These approximations are useful tools in the generation of control policies when the optimal value function is not available. If  $V^*(b)$  can be approximated by  $\tilde{V}(b)$ , the corresponding policy  $\tilde{\mu}(b)$  satisfies

$$\tilde{\mu}(b) = \arg \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \sum_{\theta \in \Theta} P(\theta|b, X) \tilde{V}(\varphi(b, \theta, X)) \right\}. \quad (2.9)$$

There are several ways of approximating  $\tilde{V}(b)$ . Bertsekas [9] discusses traditional approaches, and Bertsekas and Tsitsiklis [11] suggest recently developed methods based on neural networks and simulations. This section gives an example of two approximation methods that can be used to generate policies.

### Myopic Policy

The myopic policy consists of approximating the value function with a constant (which can be zero, without loss of generality). Therefore, the myopic policy consists of maximizing immediate payoffs

$$\mu^M(b) = \arg \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) \right\}.$$

Simple as it may be, the myopic policy is frequently used in practice. It tends to perform well with the prior belief is near the corners of the belief space simplex. It is also asymptotically optimal when there are no actions that evoke equal reactions for different states (c.f. [102]).

### n-Step Lookahead

The n-step Lookahead policy consists of approximating the optimal value function  $V^*(b)$  with the value function  $V^n(b)$ , obtained after  $n$  applications of the dynamic programming

mapping (2.6). The  $n$ -step lookahead policy

$$\mu^{nLA} : \mathbf{B} \rightarrow \mathbf{X}$$

can be obtained by substituting  $V^n(b)$  for  $\tilde{V}(b)$  in (2.9), yielding:

$$\mu^{nLA} = \arg \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \left[ \sum_{\theta \in \Theta} P(\theta|b, X) V^n(\varphi(b, \theta, X)) \right] \right\}$$

One important special case of the  $n$ -step lookahead policy is the 1-step lookahead policy  $\mu^{1LA}$ , defined by

$$\mu^{1LA} = \arg \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \left[ \sum_{\theta \in \Theta} P(\theta|b, X) \tilde{R}^*(\varphi(b, \theta, X)) \right] \right\}$$

where

$$\tilde{R}^*(b) = \max_{X \in \mathbf{X}} \left[ \tilde{R}(b, X) \right].$$

The theorem below shows how to estimate the difference between  $V^*(b)$  and  $V^n(b)$ .

**Theorem 13** *If  $V^k(b)$  is the value function obtained with the  $k$ -step lookahead policy and  $V^*(b)$  is the optimal value function, then*

$$\sup_{b \in \mathbf{B}} |V^k(b) - V^*(b)| \leq \frac{2\varepsilon\gamma^k}{1-\gamma}$$

where

$$\varepsilon = \sup_{b \in \mathbf{B}} |V^k(b) - V^{k-1}(b)|.$$

**Proof.** This is a standard result in infinite-horizon dynamic programming, and proofs can be found in textbooks such as [9]. ■

### Perfect Information Upper Bound

The policy derived from the perfect information value function approximation is given by

$$\mu^{PI} = \arg \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \sum_{\theta \in \Theta} P(\theta|b, X) V^{PI}(\varphi(b, \theta, X)) \right\}. \quad (2.10)$$

The value function  $V^{PI}(b)$  is a vector with  $|\mathbf{S}|$  components. Each component  $V_i^{PI}(b)$  is the solution of the fully observable Markov decision problem when all components of the belief vector are zeros except for the  $i$ 'th component, which is equal to 1. Astrom [?] and White [97] explain in detail how this problems are constructed and why they provide an upper bound for the POMDP.



The value function  $V^{PI}(b)$  can then be substituted into (2.10) to generate a control policy. Lovejoy [63] invokes a theorem from Van Hee [94] and notes that the bound can be tightened by applying the operator  $T$  (3.13):

$$TV^{PI}(b) \leq V^{PI}(b) \forall b.$$

Since the operator  $T$  is a contraction mapping, its repeated application generates an increasingly better sequence of upper bounds until it converges to a fixed point solution:

$$\begin{aligned} T^2V^{PI}(b) &\leq TV^{PI}(b) \forall b, \\ &\vdots \\ T^nV^{PI}(b) &\leq T^{n-1}V^{PI}(b) \forall b \\ &\vdots \\ V^{PI*}(b) &= TV^{PI*}(b). \end{aligned}$$

Solving the fixed point equation yields

$$V^{PI*}(b) = b \cdot \max_{X \in \mathbf{X}} \frac{p_i^A(X_i)}{1 - \gamma(1 - p_i^L(X_i))} \geq V^*(b)$$

## Other Approximations

As mentioned at the beginning of this section, there are too many approximation methods that could be mentioned. Just about any paper that describes a POMDP implementation will contain an approximation of the value function based on characteristics of the specific problem. Hauskrecht [44] provides a review of some approximations, as do Littman et al. [59] in the wider context of reinforcement learning. White [100], White and Scherer [101], and Lovejoy ([60], [61] and [62]) also contain descriptions of approximate solution methods.

## 2.5 New Approximation Method

### 2.5.1 Approximating the Convex Hull

This section addresses the issue of generating control policies by approximating the convex hull generated by the  $\alpha$  vectors. This is a new approximation method, and can be used to solve any POMDP. The general idea is to approximate the sets  $A^n$  with sets  $\tilde{A}^n \subset A^n$  that can be constructed efficiently. These sets can then be used to generate control policies that lower-bound the value function, since

$$\tilde{V}^n(b) = \max_{\alpha \in \tilde{A}^n} (b \cdot \alpha) \leq \max_{\alpha \in A^n} (b \cdot \alpha) = V^n(b)$$

as long as  $\tilde{A}^n \subset A^n$ .

The set of vectors  $A^n$  is generated from  $A^{n-1}$  according to the formula

$$A^n = \bigcup_{X \in \mathbf{X}} \left\{ \bigcup_{\alpha^{\theta_1} \in A^{n-1}} \left[ \bigcup_{\alpha^{\theta_2} \in A^{n-1}} \dots \left( \bigcup_{\alpha^{\theta_i} \in A^{n-1}} \left[ R(X) + \gamma \sum_{\theta_i \in \Theta} \Gamma^{\theta_i}(X) \cdot \alpha^{\theta_i} \right] \right) \right] \right\}$$

where  $R(X)$  is the  $|\mathbf{S}|$ -dimensional vector whose components are  $R(X, S_i)$ . The equation above implies that each vector in  $A^{n-1}$  is used  $(|\mathbf{X}| \cdot |A^{n-1}|)$  times in the generation of  $A^n$ . Figure 2-11 shows an example where  $|\Theta| = 2$ ,  $|\mathbf{X}| = 4$ ,  $|A^{n-1}| = 4$ , and there are only two possible observations, i.e.,

$$A^n = \bigcup_{X \in \mathbf{X}} \left\{ \bigcup_{\alpha^{\theta_1} \in A^{n-1}} \left[ \bigcup_{\alpha^{\theta_2} \in A^{n-1}} \left( R(X) + \gamma \Gamma^{\theta_1}(X) \cdot \alpha^{\theta_1} + \gamma \Gamma^{\theta_2}(X) \cdot \alpha^{\theta_2} \right) \right] \right\}$$

The graph in Figure 2 – 11a shows all 16 points that can be generated, and the graph in Figure 2 – 11b shows how the update process can be approximated by choosing the 4 circled points, reducing the complexity of the update rule from

$$|A^n| = |\mathbf{X}| |A^{n-1}|^2.$$

to

$$|\tilde{A}^n| = |\mathbf{X}| |\tilde{A}^{n-1}|.$$

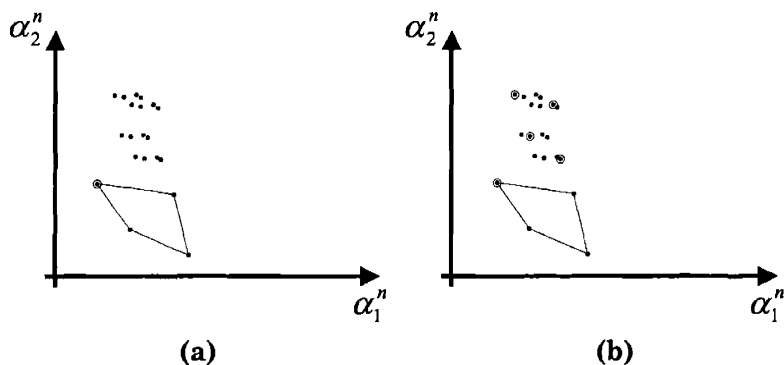


Figure 2-11: Reducing the complexity of the update rule

Figure 2-12 reproduces the sets from Figure 2-6 and adds the sets  $|\tilde{A}^n|$  below them for  $n=1, 2, 3$ .

There are  $\binom{|A^{n-1}|}{|\mathbf{X}|}$  ways to choose  $|\mathbf{X}|$  vectors from a set of size  $|A^{n-1}|$ , and all of them lead to lower bounds. In this section we consider one rule which is particularly appealing

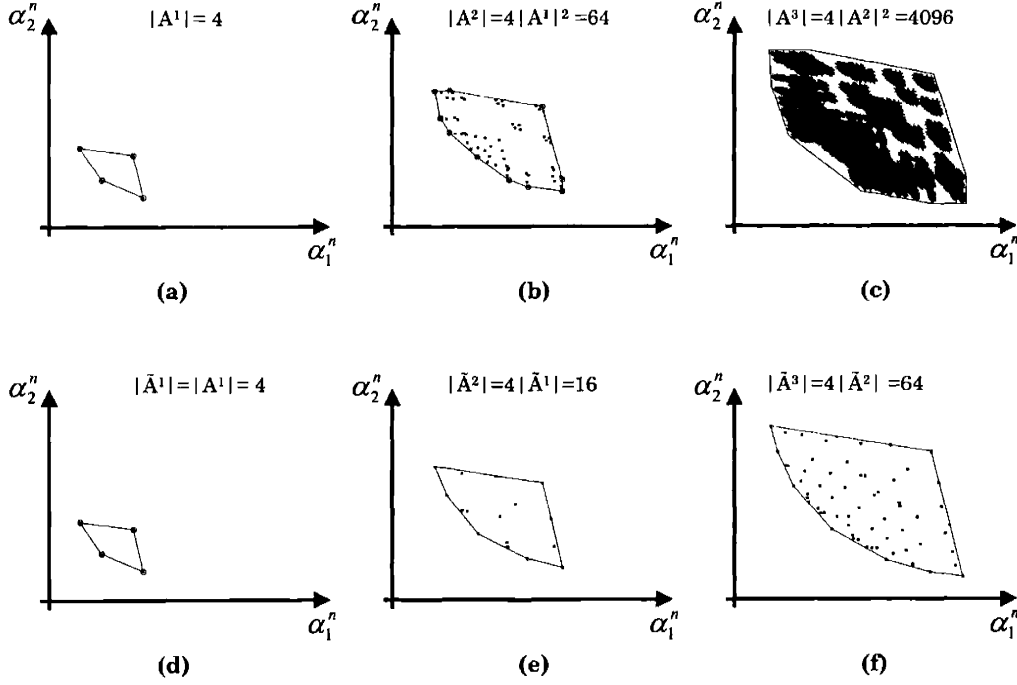


Figure 2-12: Comparing actual sets  $A^n$  with their approximations  $\tilde{A}^n$

because its corresponding control policy can be described in a closed-form solution. This rule for generating sets  $\tilde{A}^n$  consists of considering only pairs of vectors  $\alpha^i, \alpha^j \in \tilde{A}^{n-1}$  such that  $\alpha^i = \alpha^j$ . If sets  $\tilde{A}^n$  are created in this manner, we have

$$\begin{aligned}
\tilde{A}^n &= \left\{ \bigcup_{X \in \mathbf{X}} \left\{ \bigcup_{\alpha^i \in \mathbf{A}^{n-1}} \left[ \bigcup_{\alpha^j \in \mathbf{A}^{n-1}} (R(X) + \gamma \Gamma^{\theta_1}(X) \cdot \alpha^i + \gamma \Gamma^{\theta_2}(X) \cdot \alpha^j) \right] \right\} \mid \alpha^i = \alpha^j \right\} \\
&= \bigcup_{X \in \mathbf{X}} \left\{ \bigcup_{\alpha^i \in \mathbf{A}^{n-1}} (R(X) + \gamma \Gamma^A(X) \cdot \alpha^i + \gamma \Gamma^R(X) \cdot \alpha^i) \right\}. \\
&= \bigcup_{X \in \mathbf{X}} \left\{ \bigcup_{\alpha^i \in \mathbf{A}^{n-1}} (R(X) + \Gamma(X) \cdot \alpha^i) \right\}
\end{aligned}$$

where  $\Gamma(X) = \gamma [\Gamma^{\theta_1}(X) + \Gamma^{\theta_2}(X)]$ .

## 2.5.2 Numerical Studies

This section presents the results of preliminary numerical experiments aimed at investigating the suitability of the algorithm presented in this section for the solution of POMDPs. These simulations were done by randomly generating a set of parameters within a given framework (see the Appendix for details on the parameters) and simulating repeated in-

interactions between the controller and the environment using different control policies. The problems used for this experiment were partially observable stochastic shortest path problems, which ensure the existence of a trapping state and the termination of each trial in a finite amount of time. Five hundred trials were performed for each set of parameters.

Figure 2-13a shows the value functions obtained for one of the sets of parameters. The value function obtained by controlling according to the myopic policy,  $V^M(b)$  performs  $V^{\bar{A}}(b)$ . This property is more clearly observed in Figure 2-13b, which shows  $\frac{V^{\bar{A}}(b)-V^M(b)}{V^M(b)}$  as a function of the belief  $b$ .

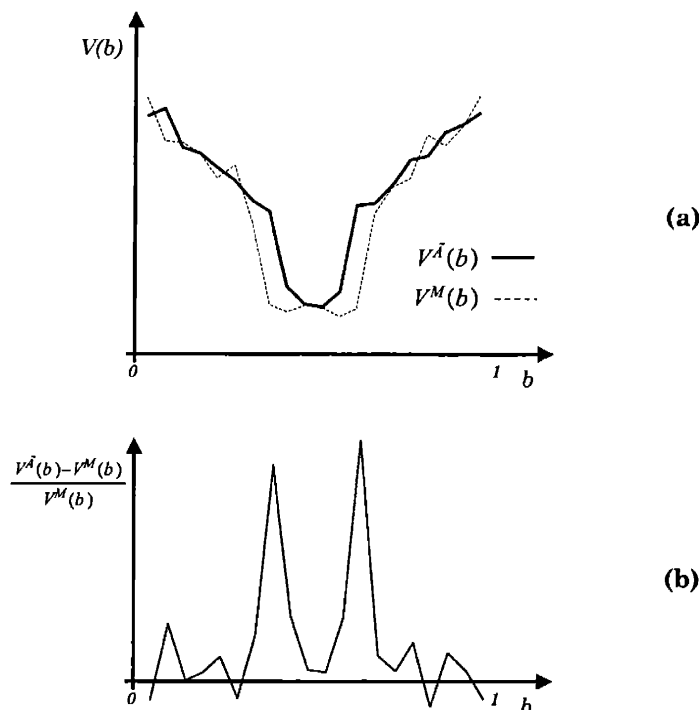


Figure 2-13: Empirical comparison of the myopic policy  $V^M(b)$  and the approximate policy  $V^{\bar{A}}(b)$

Figure 2-14 presents the results of the 16 simulations for 20 different priors. Five hundred runs were performed for each prior for each set of parameters. The columns correspond the different priors, and the rows to the set of parameters used. The results presented are the percentage difference in value between the two policies, i.e.,  $100 \frac{V^{\bar{A}}(b)-V^M(b)}{V^M(b)}$ . It is clear from these results that the pattern depicted in 2-13 is typical, albeit in more or less pronounced form in some cases. Note, for example, that the values in the columns 0.375 and 0.675 are always positive and large (around 30% or so). As knowledge approaches certainty (i.e., as the belief approaches 0 or 1) the two policies perform equally well. The percentage differences can be positive or negative, but are always very small.

The optimal solution was calculated for a few small problems and the value function

	0.025	0.075	0.125	0.175	0.225	0.275	0.325	0.375	0.425	0.475	0.525	0.575	0.625	0.675	0.725	0.775	0.825	0.875	0.925	0.975
1	-5.627	0.481	2.989	-1.101	7.079	8.868	-11.085	54.990	-0.575	-0.889	-4.663	5.368	32.095	-0.860	3.038	2.593	-3.474	-14.548	-9.630	-7.257
2	-2.795	3.064	2.189	-0.487	6.727	-2.539	-2.036	23.795	5.502	8.105	-2.749	6.554	25.785	0.572	0.118	6.591	-10.918	2.806	11.989	-6.579
3	-9.858	10.900	1.426	-1.173	-2.147	-6.255	15.317	11.391	-0.406	-4.982	5.890	6.661	27.291	-0.427	3.955	1.432	-11.592	8.894	3.706	-0.075
4	-5.734	10.065	7.010	3.532	3.962	-2.588	6.922	9.450	11.846	-9.224	-1.374	7.482	27.461	-0.188	-5.057	-4.237	-8.078	3.594	10.890	-1.903
5	-5.357	6.287	-1.176	7.397	2.802	-5.779	0.148	20.480	20.791	9.199	0.998	8.940	33.842	5.098	-0.637	21.706	0.652	-1.010	1.754	-0.548
6	-2.063	11.822	-3.386	-7.733	3.372	7.659	12.004	29.774	8.283	4.399	-6.264	0.895	37.395	0.090	-0.835	-1.889	11.203	12.123	-7.431	-6.848
7	0.276	7.837	-11.308	3.631	-1.613	-9.313	11.398	24.028	0.038	-6.124	10.301	3.538	34.423	3.679	-0.776	6.757	-10.000	1.123	-7.780	0.827
8	4.805	0.259	-3.888	-3.240	-3.023	-12.971	9.070	39.166	16.022	4.880	-0.075	17.347	16.580	10.280	2.956	-0.264	0.886	8.867	2.582	-2.815
9	-3.647	4.654	4.375	-1.514	-4.109	7.566	-0.649	23.841	4.296	1.740	-4.344	1.136	44.590	13.302	2.442	-5.438	-0.748	0.127	-6.799	2.124
10	-10.359	2.477	4.323	4.953	15.364	-8.665	2.530	34.323	5.211	-5.056	6.991	8.986	26.444	-0.361	-2.179	7.816	-9.445	8.220	18.454	-8.577
11	-0.256	-3.179	-0.565	-5.022	-2.938	-4.905	3.610	30.362	2.103	8.003	1.689	9.302	45.295	-1.555	3.233	-4.155	-9.264	1.848	-0.803	-1.931
12	3.600	8.985	8.806	4.217	-2.053	-7.489	14.462	14.497	6.572	-5.491	-5.015	10.499	29.931	1.098	-9.021	-4.887	-5.822	-1.742	9.662	-11.762
13	-4.011	16.709	2.754	-5.367	-1.976	4.019	-7.022	29.490	24.702	7.645	4.281	10.159	28.208	10.432	4.821	14.674	7.444	1.328	1.836	-4.058
14	-0.025	19.828	-2.585	4.102	4.726	-3.257	9.079	28.464	8.243	1.534	-5.073	-2.412	23.160	-0.750	-3.996	3.445	4.647	5.491	-3.422	-0.797
15	-0.877	4.845	-8.110	-0.468	-0.293	-10.703	7.514	25.019	-1.809	-2.502	10.417	2.399	26.731	-5.214	3.507	2.291	-6.925	7.204	-6.543	-0.150
16	-10.547	-2.562	-9.669	1.051	8.899	0.660	11.610	23.791	11.860	-0.610	-6.624	16.917	11.800	3.323	4.314	19.032	-11.142	-1.125	-5.689	-0.328

Figure 2-14: Simulation Results. Columns are initial belief levels (priors). Rows are different trials (parameter values). Entries are  $\frac{V^{\hat{A}}(b) - V^M(b)}{V^M(b)}$ .

obtained through its numerical implementation was found to be statistically indistinguishable from the approximate value function described in this section. However, other approximation methods such as 10-step lookahead also performed equally well. The the new approximation method has a theoretical appeal due to the manner in which it is constructe, building indirectly on Sondik’s [86] result. Morevoer, the simulations performed so far suggest that this method may also be appealing from an empirical perspective. However, a more precise characterization of the circumstances under which the present method outperforms currently existing approximations remains an open question and is the subject of current investigation.

## 2.6 Conclusions

POMDPs provide a rich analytical framework for a wide variety of problems in many different fields. In particular, there is great opportunity for applying POMDPs in Management Science. The advent of the internet and the possibility to model human/computer interactions as POMDPs makes these problems more relevant then ever. To add to this momentum, the recent interest in POMDPs from the AI community has generated a fertile ground for new advances.

As always, the major obstacle to the widescale application of POMDPs continues to be the lack of efficient exact solution methods. It seems to be the case that the computation of optimal policies will continue to be impractical for most problems. Nevertheless, is important to keep searching for exact solution methods to inspire sub-optimal solution methods. History has shown that even though exact solution methods are rarely adequate to implement real-world problems they often provide a framework for the development of suboptimal control policies.

## 2.7 Appendix

### 2.7.1 Parameter Values

This appendix describes the process for generating the parameters for numerical studies in §2.5.2. This problem is similar to the customer/company interactions described in the next chapter. Please refer to that chapter for an interpretation of the meaning of these parameters.

The state space consists of two customer types:  $\mathbf{S} = \{(\beta_{11} \ \beta_{12}), (\beta_{21} \ \beta_{22})\}$ . The values for the parameters  $\beta$  were chosen randomly from a uniform distribution over the interval  $[-3, 3]$ . The controls were  $\mathbf{X} = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$ . The possible observations were Accepting, Rejecting, and Leaving. The probabilities with which these observations occurred are based on a random utility model as described in the next chapter. The revenues are also as described in the next chapter: 1 if the customer accepts the recommendation, and 0 otherwise.

### 2.7.2 The Exploration/Exploitation tradeoff

The fundamental difficulty in solving POMDPs lies in the fact that the goal of maximizing learning often stands in conflict with the goal of maximizing immediate payoffs. This conflict is referred to as the “exploration/exploitation tradeoff” in the Artificial Intelligence (AI) literature. The exploration/exploitation tradeoff lies at the heart of a number of problems in AI, and has therefore received a significant amount of attention in recent years. The field of Reinforcement Learning (also referred to as Machine Learning) has developed a number of different techniques to control systems facing the exploration/exploitation tradeoff. Reinforcement learning is defined in [48] as “the problem faced by an agent that learns behavior through trial-and-error interactions with a dynamic environment”. Given this general definition, a comprehensive review of all the methods of reinforcement learning is beyond the scope of this dissertation. The interested reader is referred to [48] as well as the textbooks by Sutton and Barto [88] and Mitchell [68] for an introduction to this vast field. In this appendix, we briefly review some recent developments in the AI literature that address the exploration/exploitation tradeoff within the framework of POMDPs. These papers are not directly relevant to this dissertation in that our focus is on the mathematical structure of POMDPs and their interpretation in the context of managerial decision-making. However, these papers are indirectly relevant in that good POMDP approximation methods could prove to be useful in a large-scale implementation of the model developed in the next two chapters.

One popular approach to implementing POMDP models has been to compute a belief-state value function offline and keep track of the current belief online in order to implement the control policy. The value function is typically computed using approximation methods such as those described in §2.4.4. However, an additional problem in large-scale implementations is estimating the current belief. In order to solve this problem, a new class of

approximation methods using factored representations has been recently proposed. Boutilier and Poole [16] showed how factored representations, which are common in other areas of artificial intelligence, can be integrated into the POMDP framework in such a way that allows for state abstraction and simplifies computations during the implementation stage. Hansen and Feng [42] devised new methods of implementing the ideas from [16] that outperform the original implementation. Poupart and Boutilier [74] propose a new way of approximating the belief state, where approximation quality is determined by the expected error in the utility function (as opposed to an absolute error in the estimation of the belief itself). More recently, Poupart and Boutilier [75] devise new search procedures for selecting approximation schemes. These methods use a vector space analysis of the problem, and build on the authors' previous value-directed methods ([74]), but are significantly more efficient from a computational perspective (up to two orders of magnitude faster).

Another important recent development is the use of hierarchical learning methods in the implementation of control policies in stochastic domains. This line of research builds upon the findings of Precup and Sutton [76] and Precup, Sutton and Singh [77] in the realm of temporally abstract planning. Hauskrecht et al [45] derived control policies for an MDP using macro-actions (from an abstract MDP) that use a reduced state space. Kaelbling applied similar techniques to stochastic domains in [46]. She did so by developing the HDG algorithm, which generalized her DG algorithm [47] which is analogous to Watkins' Q-learning ([95], [96]).

## Chapter 3

# Design to Learn: Customizing Services when the Future Matters

### 3.1 Introduction

The Internet presents unprecedented opportunities for the customization of services. Companies can identify returning customers at almost no cost and present each of them with a unique interface or service offering over a web page, voice interface, or wireless device. Customization is important in online interactions because the possibilities of products and services that companies can offer can be enormous and customers are often not able to find and identify the alternative that best suits their preferences. More generally, the amount of information available over the Internet is overwhelming and the costs to search and evaluate all potentially relevant pieces of information can be prohibitive. This situation can lead to interactions of low net utility to customers. Hence the need for software applications generally called intelligent agents (or smart agents) that present customers with customized recommendations from large sets of alternatives. These agents have the potential to create great benefits for both consumers and companies. It is certainly desirable for customers to reduce search costs and by going directly to a company that already knows them and gives them high utility. At the same time, companies that learn about their customers can use this knowledge to provide increasingly better service as the relationship progresses, making it difficult for competitors to “steal” their customers. In order to fully reap the benefits of customization, companies must overcome two obstacles. First, the strategies that maximize their customers’ utility (or probability of sale in the current period) do not coincide with strategies that maximize learning. Second, customers rarely forgive and return to smart agents that make bad recommendations. Overcoming these obstacles depends on the agents’ ability to adequately balance the goals of learning at the expense of selling and selling at the expense of learning.

The primary function of agents is to help customers make better decisions by reducing search costs and making recommendations of products which present characteristics that are difficult to describe and evaluate. Travel agents, for example, save their customers time



by looking through all the different tickets with all their restrictions and identifying the cheapest fare that will satisfactorily fulfill the customers' needs. Real estate agents not only have access to a larger number of available houses than most customers do, but they can also help househunters make their decisions by teaching them about important aspects of evaluating a house before the purchase. On the Internet, smart agents can create customized newspapers by searching enormous databases of news articles and selecting the best ones for each individual customer. Or they can decide which songs to play in a web-based radio station individually tailored for each user.

Agents are playing an increasingly important role in consumers' lives because of the Internet. The overwhelming amount of information available through the Internet has generated a high demand for agents that reduce search costs. In addition to being an important necessity, smart agents are also cost-efficient. Consequently, smart agents play a much larger role in everyday decision-making and have considerable more autonomy to act on the consumers' behalf [64]. Few consumers have human agents who choose newspaper articles, but these types of services are feasible and increasingly popular over the Internet.

The essential task of learning about customers in order to provide better recommendations has been made more difficult by the migration of customer interfaces from human-human to human-computer. Human agents can, in many cases, observe their clients and make a number of inferences on the basis of their interactions. The lack of cues typical of the personal encounter diminishes the opportunities for correcting incomplete advice or wrong guesses. Software agents only register click patterns and browsing behavior. Smart agents can only learn about customers by asking them questions and by observing their behavior. The issues related to the benefits and limitations of asking questions have been adequately addressed elsewhere: clients may not be willing nor have the patience to answer questions [92], and their replies are not always truthful for various (and sometimes intentional) reasons [98]. Regardless of how they answer surveys, customers spend their time and money in the activities from which they derive the most utility. In the offline world, Rossi et al. [82] found that the effectiveness of a direct marketing campaign was increased by 56% due to the customization of coupons based on observations of one single purchase occasion. In web-based settings, learning can occur regardless of whether or not customers are satisfied with the products recommended to them. Therefore, companies must strive to learn about their customers by observing how they react to recommendations.

Intelligent agents face a dilemma: they must either sell at the expense of learning or learn at the expense of selling. In other words, they can either make recommendations that they expect their customers to like and learn about their customers' preferences at slow rates or they can take more risks by suggesting products their customers might not accept and learn at higher rates. The agent's dilemma captures the essence of the selling versus learning trade-off faced by companies that customize services on the Internet. Observing an action that was already expected to happen only reconfirms something one already knew or suspected to be true. But should an agent recommend a product with no prior knowledge of how their customers would react? By taking this course of action, agents may learn a lot

about their customers' responses to recommendations of such products, regardless of whether or not the item is purchased. However, they run the risk of causing an unsuccessful service encounter that may lead to defection.

Customers who receive very bad recommendations can lose trust in the smart agent's ability to help them make better decisions and never return. Research in psychology and marketing has repeatedly shown the importance of trust as the basis for building effective relationships with customers in e-commerce as well as in regular commerce (e.g., [52], [55], [4]). Doney and Cannon [27] established an important link between trust and profitability by noting the importance of trust to build loyalty. Low trust leads to low rates of purchase, low customer retention and, consequently, meager profits. Failure to sell has been equated with failure to build trust ([55], [71]). Studies on human and non-human agents have extended the findings of previous research by showing that trustworthy agents are more effective, i.e., they are better at engaging in informative conversation and helping clients make better decisions. Urban et al [93] have also shown that advice is mostly valued by less knowledgeable and less confident buyers, and those are exactly the same customers less likely to trust external agents to act as surrogates on their behalf.

Intelligent agents must strive to be trustworthy in two ways. First, customers must trust that agents have their best interest in mind. This will be true whenever the agent's payoffs are proportional to each customer's utility. Second, the customer must trust that the agent is good. Models of online consumer behavior must take into account the fact that if the agent makes a very bad recommendation the consumer loses trust and may never come back. The consequences of giving bad advice are much worse for software agents than for human agents. Ariely [4] conducted a series of experiments to compare trust online and offline and found that even though customers are sometimes willing to forgive a mistake made by a human agent, they rarely do so for a software agent. In one of his experiments, half of the subjects received financial advice from a software agent, and the other half from a human agent. Both agents were manipulated so that they made exactly the same serious mistake. After realizing they had received bad advice, customers of the human agent were much more likely to continue the relationship than customers of the software agent. People forgive people but they don't forgive software. Turning off the computer or clicking on a different website entails much less psychological costs than terminating a personal relationship. Consequently software agents face much higher churn rates than their human counterparts. The levels of tolerance for incomplete, wrong or misleading advice are lower, yet the electronic agent must rely on much more precarious information and cues regarding consumers' tastes and preferences.

The agent's dilemma is related to a number of problems where decision-makers must decide whether to exploit their knowledge of the system or to explore in order to gain more knowledge to improve future payoffs. The most famous problem of this type is the multi-armed bandit ([37],[38]), where the decision maker sequentially selects one of  $n$  different independent stochastic processes ("arms") for observation. The independence assumption is crucial to the solution proposed by Gittins [37] and unacceptable in the problem studied in

this paper. In many circumstances, agents can learn about a customer's preferences about product  $B$  by observing his or her reaction to product  $A$ . There have been efforts to allow for dependent arms in the bandit problem by introducing covariates ([24], [39]) but there is no general solution method. Furthermore, none of the methods proposed in the literature account for the fact that the process can be terminated at any time depending on the outcome of the process, which corresponds to the customer leaving in this paper's application.

The framework of partially observable Markov decision processes (POMDPs) (e.g.: [29], [70], [19]) can be used to model the agent's decision problem as formulated in this paper. §3.2 shows how the agent's dilemma can fit into the framework of POMDPs. §3 establishes the optimality conditions and basic solution procedures for the model introduced in §3.2. §3.4 discusses how the optimal solution can be computed through numerical methods and §3.5 shows how good recommendation policies can be constructed through value function approximations or simplifications of the model. §3.6 discusses the significance of the results of this paper in the managerial and academic contexts. Finally, §3.7 offers some directions for further research.

## 3.2 Model Description

This section is divided into four parts. The first one presents a mathematical choice model that captures the essential behavioral features of the choice process customers go through when interacting with software agents. This is followed by a description of a dynamic system used to model the interactions between customers and companies over time. The third section shows how the company's decision of how to customize products and services can be framed in terms of a Markov Decision Process using the customer behavior model of §3.2.1 and the dynamical system of §3.2.2. Finally, §3.2.4 summarizes the main features of the model before proceeding to the analysis of the optimization problem.

### 3.2.1 Customer Behavior

This section describes a customer behavior model that realistically captures the ways in which customers react to recommendations of products and services made by software agents. Such model is an essential step in the quantification of the costs and benefits of customization from the company's perspective. Customers are satisfied when they accept the suggested product or service, and return to the same company the next time they need a service from the same industry. If a recommendation is barely below the acceptance level, customers will purchase elsewhere (or forego purchasing in the current period, as the case may be) but return the next time they need recommendations. In this case, companies incur the cost of losing a sale opportunity. If the utility of the suggested product is perceived to be significantly below the acceptable level, customers infer that the agent is bad and unable to meet their needs. Bad agents are not trustworthy, and customers will never return. The likelihood that the

customer will accept the recommendation, reject the recommendation, or leave the company are defined through a random utility model.

Random utility models were introduced in the psychology literature by Thurstone [91], mathematically formalized by McFadden [66] and recently reviewed by Meyer and Kahn [67]. According to random utility theory, when a customer is offered product  $X$  the utility that is actually observed has two components: a deterministic component  $u(X)$ , which corresponds to the true underlying utility and an error term  $\varepsilon$ . The error term accounts for factors such as unobserved attributes, imperfect information, or measurement errors make the customer unable to determine the exact utility of a product upon initial examination. Formally, this can be expressed as

$$\hat{u}(X) = u(X) + \varepsilon$$

where  $\hat{u}(X)$  is the observed utility.

Once customers evaluate the utility of  $X$ , the service they are being offered, they can take one of three actions:

- Accept product  $X$  if  $u(X) + \varepsilon > u(\lambda)$
- Reject product but come back next time if  $-c < u(X) + \varepsilon < u(\lambda)$
- Leave the company and never come back if  $u(X) + \varepsilon < -c$

$u(\lambda)$  is the minimum utility required for the customer to accept the product. Without loss of generality, it is assumed throughout this paper that  $u(\lambda) = 0$ , since this simply amounts to scaling utilities.

The probabilities of accepting, leaving, or rejecting a product  $X$  (denoted  $p^A(X)$ ,  $p^L(X)$ , and  $p^R(X)$ , respectively) can be computed by defining the distribution of  $\varepsilon$ . One useful technique is to assume that  $\varepsilon$  is normally distributed with mean 0 and variance  $\sigma^2$ . The parameter  $\sigma$  can be altered to represent different types of products or business settings, where the customer observes products with more or less error. This model corresponds to the ordered Probit model, and its properties are thoroughly discussed in Greene [40]. The general formulation of the choice probabilities is shown in the first column of (3.1), and the specific form they take when  $\varepsilon \sim N(0, \sigma)$  is shown in the second column. Figure 3-1 depicts the probabilities taking each of the three possible actions when the customer is offered a recommendation with a deterministic utility component  $u$ . In this particular case,  $u$  is below the acceptance threshold, but since the observed utility is  $u + \varepsilon$  the probability that the product will be accepted is positive.

Action	Probability of Action	Probability if $\varepsilon \sim N(0, \sigma)$
Accept	$p^A(X) = \Pr(u(X) \geq 0)$	$= \Phi\left(\frac{u(X)}{\sigma}\right)$
Leave	$p^L(X) = \Pr(u(X) < -c)$	$= \Phi\left(-\frac{u(X)+c}{\sigma}\right)$
Reject	$p^R(X) = [1 - p^A(X) - p^L(X)]$	$= 1 - \Phi\left(\frac{u(X)}{\sigma}\right) - \Phi\left(-\frac{u(X)+c}{\sigma}\right)$

(3.1)

The choice model of (3.1) captures customers' reactions to recommendations made by software agents in a way that is consistent with the behavior qualitatively described in the psychology and marketing literature (e.g., [4]). Customers who observe a very low utility ( $-c$ , in our notation) immediately lose trust in the agent as a reliable advisor and never return. The recommendation will only be accepted (i.e., the product will only be purchased) if the observed utility exceeds 0. Finally, the cost of not making a sale by experimenting with different customization policies is captured by the probability that the observed utility will fall in the interval  $[-c, 0]$ .

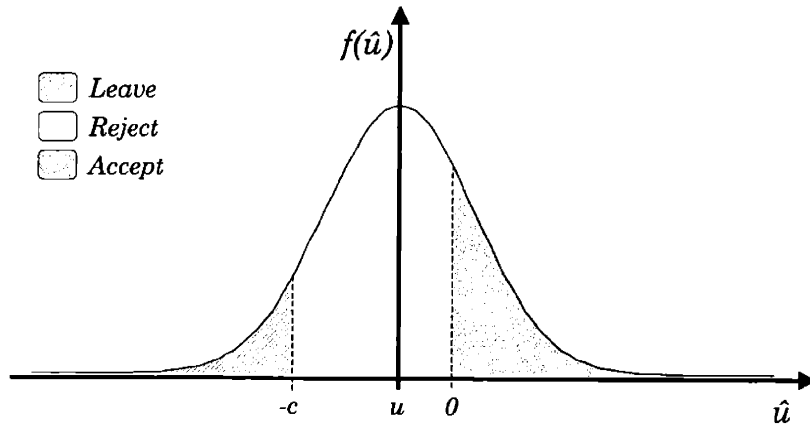


Figure 3-1: Probabilities of Accepting, Rejecting, and Leaving when offered a product with deterministic utility  $u$

### 3.2.2 Dynamics of Customer-Company Interactions

Figure 3-2 describes the dynamics of customer/company interactions. This system is the basis of the formulation of the agent's dilemma as a Markov decision problem.

A new customer arrives from a population consisting of  $|\mathbf{S}|$  different segments. Each customer belongs to a segment  $S_i \in \mathbf{S}$ . The company does not know the segment to which the new customer belongs, but has prior beliefs. These beliefs are represented by a  $|\mathbf{S}|$ -dimensional vector  $b^0 \in \mathbf{B}$ , with each component corresponding to the probability that the customer belongs to a given segment. More specifically, if  $S^*$  denotes the true segment to

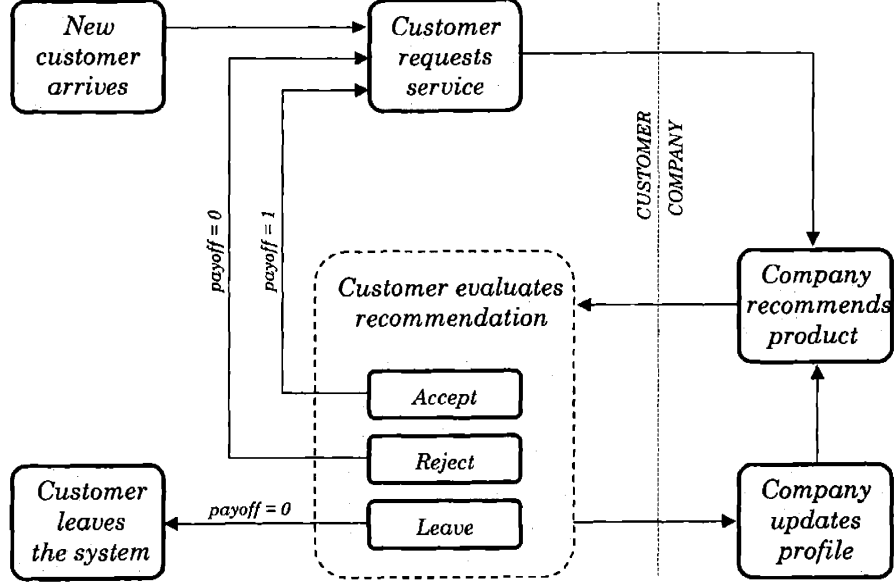


Figure 3-2: The dynamics of customer-company interactions

which the customer belongs the prior beliefs are

$$b^0 = \begin{pmatrix} \Pr(S^* = S_1) \\ \Pr(S^* = S_2) \\ \dots \\ \Pr(S^* = S_{|S|}) \end{pmatrix} := \begin{pmatrix} b_1^0 \\ b_2^0 \\ \dots \\ b_{|S|}^0 \end{pmatrix}$$

where  $b_i^0 \in [0, 1] \forall i$  and  $\sum_{i=1}^{|S|} b_i^0 = 1$ .

Each segment  $S_i$  is defined by a utility function  $u_i(X)$  that maps products and services  $X$  into real numbers

$$u : \mathbf{S} \times \mathbf{X} \rightarrow \mathbb{R}$$

One way to define these utility functions is by associating each segment  $S_i$  with a vector  $\beta_i$ . The components of  $\beta_i$  are the weights a customer of segment  $i$  gives to each attribute of the products in  $\mathbf{X}$ . In this case, the utility function could be defined as

$$u_i(X_j) = (\beta_i' \cdot X_j)$$

where  $u_i(X_j)$  denotes the utility that product  $X_j$  has to a customer of segment  $S_i$ .

When the customer requests service, the company chooses a product from the set  $\mathbf{X} = \{X_1, X_2, \dots, X_{|\mathbf{X}|}\}$ .  $\mathbf{X}$  is a set of substitute products or services, and can be thought of as the different ways in which a service can be customized. Each  $X_i \in \mathbf{X}$  is a vector whose components correspond to a different attribute of the product or service. The recommenda-

tion made during the  $t$ 'th interaction is denoted  $X^t$ . The decision problem of which service configuration should be chosen is directly addressed as an optimization problem in § 3.2.3.

The customer will either accept the recommendation, reject the recommendation, or leave, as explained in the beginning of this section. The probabilities with which each of these actions take place are given by (3.1). The company will observe the customer's action as described in Figure 3-2. The observation made during the  $t$ 'th interaction is denoted  $\theta^t$ , and the set of possible observations is  $\Theta = \{\theta^A, \theta^R, \theta^L\}$ , corresponding to accepting the recommendation, rejecting the recommendation, and leaving the system. For example,  $\theta^5 = \theta^A$  means that the customer accepted the product offered during the fifth interaction. The company updates its beliefs every time it makes an observation through a function

$$\varphi : \mathbf{B} \times \mathbf{X} \times \Theta \rightarrow \mathbf{B}$$

that determines the new belief given the previous belief and the action/observation pair of the last interaction. If  $b^t$  denotes the belief vector after  $t$  interactions, then

$$b^t = \begin{pmatrix} \Pr(S^* = S_1 | b^0, X^1, \theta^1, X^2, \theta^2, \dots, X^t, \theta^t) \\ \Pr(S^* = S_2 | b^0, X^1, \theta^1, X^2, \theta^2, \dots, X^t, \theta^t) \\ \dots \\ \Pr(S^* = S_{|\mathbf{S}|} | b^0, X^1, \theta^1, X^2, \theta^2, \dots, X^t, \theta^t) \end{pmatrix} = \begin{pmatrix} b_1^t \\ b_2^t \\ \dots \\ b_{|\mathbf{S}|}^t \end{pmatrix}, \quad (3.2)$$

which can be written more compactly as

$$b^t = \begin{pmatrix} \Pr(S^* = S_1 | b^{t-1}, X^t, \theta^t) \\ \Pr(S^* = S_2 | b^{t-1}, X^t, \theta^t) \\ \dots \\ \Pr(S^* = S_{|\mathbf{S}|} | b^{t-1}, X^t, \theta^t) \end{pmatrix} = \begin{pmatrix} b_1^t \\ b_2^t \\ \dots \\ b_{|\mathbf{S}|}^t \end{pmatrix}$$

since  $b^{t-1}$  is a sufficient statistic for  $\{b^0, X^1, \theta^1, X^2, \theta^2, \dots, X^{t-1}, \theta^{t-1}\}$ , i.e.,

$$\Pr(S^* = S_i | b^0, X^1, \theta^1, X^2, \theta^2, \dots, X^t, \theta^t) = \Pr(S^* = S_i | b^{t-1}, X^t, \theta^t), \quad \forall i^1$$

The new belief can be computed in a Bayesian manner, using the relationship

$$\Pr(S^* = S_i | b^{t-1}, X^t, \theta^t) = \frac{1}{\Pr(\theta^t | b^{t-1}, X^t)} \cdot [b_i^{t-1} \cdot \Pr(\theta^t | S_i, X^t)]$$

where

$$\Pr(\theta^t | b^{t-1}, X^t) = \sum_{i=1}^{|\mathbf{S}|} [b_i^{t-1} \cdot \Pr(\theta^t | S_i, X^t)] \quad (3.3)$$

---

<sup>1</sup>Astrom [?] and Bertsekas [8] provide formal derivations of this property

is a normalizing factor. The update function  $\varphi$  can then be defined by

$$\varphi(b, \theta, X) = \frac{1}{\Pr(\theta|b, X)} \cdot \begin{pmatrix} b_1 \cdot \Pr(\theta|S_1, X) \\ b_2 \cdot \Pr(\theta|S_2, X) \\ \dots \\ b_{|S|} \cdot \Pr(\theta|S_{|S|}, X) \end{pmatrix}. \quad (3.4)$$

### 3.2.3 Optimization Problem

The company earns a payoff of 1 if the customer accepts the recommendation and 0 otherwise. Since increasing the utility always increases the probability of purchase, the incentives of the customer and the company are perfectly aligned. It is always in the company's best interest to please the customer, since the more utility the company provides to the customer, the higher the payoff. The payoff function is defined by a mapping

$$R : \mathbf{S} \times \mathbf{X} \longrightarrow \mathfrak{R},$$

and the expected payoff of suggesting product  $X_j$  to a customer from segment  $S_i$  is given by

$$\begin{aligned} R(S_i, X_j) &= 1 \cdot p_i^A(X_j) + 0 \cdot p_i^R(X_j) + 0 \cdot p_i^L(X_j) \\ &= p_i^A(X_j) \end{aligned} \quad (3.5)$$

where  $p_i^A(X_j)$  is the probability of purchase as defined in table 3.1.

If the company knows the customer's true segment, the problem of making recommendations reduces to a Stochastic Shortest Path (SSP) problem. This problem was initially formulated by Eaton and Zadeh [31] and has received a significant amount of attention in Electrical Engineering and Operations Research due to its many applications. Bertsekas and Tsitsiklis [10] and Bertsekas [9] provide an extensive review of this literature and a summary of the main results. Figure 3-3 depicts the dynamics of a problem where there are only two possible customer segments. The transition probabilities and payoffs corresponding to each arc are given as functions of the possible recommendations  $X$ . The agent's problem is to find a policy

$$\mu : \mathbf{S} \longrightarrow \mathbf{X} \quad (3.6)$$

that maps each state of the system into a product, the states in this case being the segments to which the customer can belong. The optimal policy maximizes the time to termination, i.e., the time the customer remains in the system before leaving. Technically, this is a "Stochastic Longest Path" problem, but it can be converted to a SSP by changing the payoffs from  $R$  to  $(-R)$ . The SSP terminology is used in order to stay consistent with the literature. This particular SSP has a trivial solution: always suggest the product with the highest expected utility for that segment. Unfortunately, the problem is not trivial if the company does not know the customer's segment. The company will act based on beliefs.



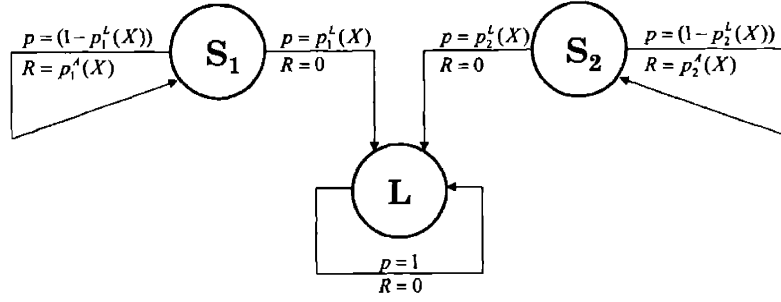


Figure 3-3: Transition probabilities and payoffs for the Fully Observable Stochastic Shortest Path problem

The decision problem faced by companies is to find a policy

$$\mu : \mathbf{B} \rightarrow \mathbf{X} \quad (3.7)$$

that determines the best product to suggest for each possible belief. This policy has the same range as the policy defined in (3.6), but it is much more complex because the domain is no longer a finite set of states: it is a set of infinitely many beliefs. In the belief-state problem, the payoff function must be map belief states and actions to revenues, i.e.,

$$\tilde{R} : \mathbf{B} \times \mathbf{X} \rightarrow \mathfrak{R}.$$

The function  $\tilde{R}$  is defined by

$$\begin{aligned} \tilde{R}(b, X) &= \sum_{i=1}^{|\mathbf{S}|} b_i \cdot R(S_i, X_j) \\ &= \sum_{i=1}^{|\mathbf{S}|} b_i \cdot p_i^A(X), \end{aligned} \quad (3.8)$$

the last equality being an immediate consequence of (3.5).

There are only three possible transitions from any given belief state,

$$\Pr(b^t = \hat{b} | b^{t-1}, X^t) = \begin{cases} \sum_{i=1}^{|\mathbf{S}|} b_i^{t-1} \cdot p_i^A(X^t) & \text{if } \hat{b} = \varphi(b^{t-1}, \theta^A, X^t) \\ \sum_{i=1}^{|\mathbf{S}|} b_i^{t-1} \cdot p_i^R(X^t) & \text{if } \hat{b} = \varphi(b^{t-1}, \theta^R, X^t) \\ \sum_{i=1}^{|\mathbf{S}|} b_i^{t-1} \cdot p_i^L(X^t) & \text{if } \hat{b} = \varphi(b^{t-1}, \theta^L, X^t) = L \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

where  $L$  corresponds to the state when the customer has left the system. The belief-state SSP problem (or Partially Observable SSP) corresponding to the problem in Figure 3-3 is depicted in Figure 3-4. The new transition probabilities are given by (3.9), and the payoffs are given by (3.8).

The customization policy that maximizes the agent's expected revenue must solve the equation

$$\mu^* = \arg \max_{X \in \mathbf{X}} \left\{ E \left( \sum_{t=0}^{\infty} \gamma^t r_t \right) \right\}$$

where  $\gamma \in (0, 1)$  is a discount factor and  $r_t$  is the reward at time  $t$ . Unlike the fully observable problem, it is no longer obvious what the optimal policy must be. The next section will derive some properties of the optimal solution. Before turning to that analysis, it is useful to summarize the results of this section.

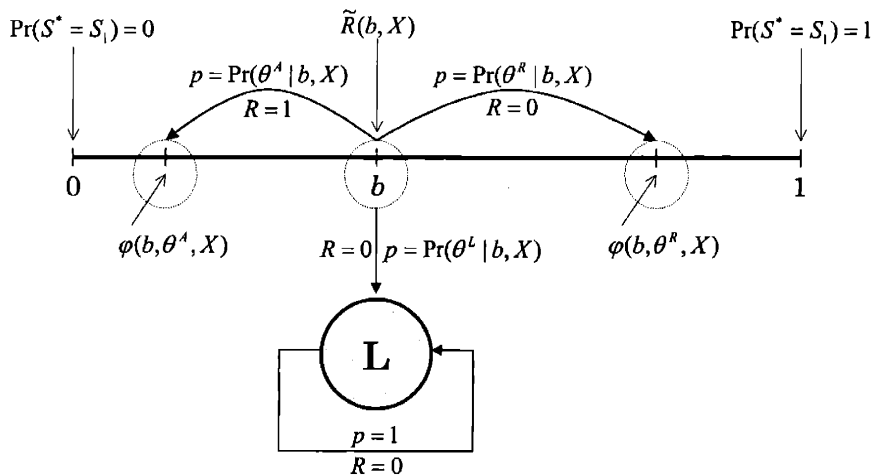


Figure 3-4: Transition probabilities and payoffs in the Partially Observable Stochastic Shortest Path problem

### 3.2.4 Summary

This section has demonstrated how the problem of finding optimal customization policies is equivalent to finding the policies satisfying the equation

$$\mu^* = \arg \max_{X \in \mathbf{X}} \left\{ E \left( \sum_{t=0}^{\infty} \gamma^t r_t \right) \right\}.$$

The dynamics of customer/company interactions are described as repeated interactions between one customer and an agent that must decide how to make a recommendation from a set  $\mathbf{X}$  based on its beliefs, as defined in (3.2). Customers observe recommendations with error,

as defined by the random utility model described in Section (3.2.1), and then decide whether to accept the recommendation, reject the recommendation, or leave. The agent observes the customer's action and receives a payoff of 1 if the customer accepts the recommendation and 0 otherwise. Finally, the agent updates its beliefs about what segment the customer belongs to according to the update function defined in (3.4). The optimization problem consists of controlling a Partially Observable Stochastic Shortest Path Problem which is a special case of the Partially Observable Markov Decision Process (POMDP) where the agent's immediate payoffs are given by (3.8) and the objective is to maximize the discounted stream of payoffs until the customer leaves the company.

### 3.3 Properties of the Value Function

#### 3.3.1 Optimality Conditions

Let  $V^*(b) = \max E \left( \sum_{t=0}^{\infty} \gamma^t r_t \right)$ . Then the optimal value function  $V^*(b)$  satisfies the Bellman equation

$$V^*(b) = \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \sum_{b' \in \mathbf{B}} P(b'|b, X) V^*(b') \right\} \quad (3.10)$$

where  $\tilde{R}(b, X)$  is the expected revenue the agent receives by suggesting product  $X$  when the belief state is  $b$ . We know from (3.9) that  $b'$  can only take three possible values, namely  $\varphi(b, \theta^A, X)$ ,  $\varphi(b, \theta^R, X)$ , and  $\varphi(b, \theta^L, X)$ . Therefore (3.10) can be written as

$$V^*(b) = \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \sum_{\theta \in \Theta} P(\theta|b, X) V^*(\varphi(b, \theta, X)) \right\},$$

which can be further simplified to

$$V^*(b) = \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma [P(\theta^A|b, X) V^*(\varphi(b, \theta^A, X)) + P(\theta^R|b, X) V^*(\varphi(b, \theta^R, X))] \right\} \quad (3.11)$$

by noting that customers have no value after they leave, and therefore  $V^*(L) = 0$ . It follows immediately that the optimal policy

$$\mu^* : \mathbf{B} \rightarrow \mathbf{X}$$

must satisfy the condition

$$\mu^*(b) = \arg \max_{X \in \mathbf{X}} \{ \tilde{R}(b, X) + \gamma [P(\theta^A | b, X) V^*(\varphi(b, \theta^A, X)) + P(\theta^R | b, X) V^*(\varphi(b, \theta^R, X))] \} \quad (3.12)$$

### 3.3.2 Value Iteration Algorithm

The DP mapping in this problem is

$$(TV)(b) = \max_{X \in \mathbf{X}} \{ \tilde{R}(b, X) + \gamma [P(\theta^A | b, X) V(\varphi(b, \theta^A, X)) + P(\theta^R | b, X) V(\varphi(b, \theta^R, X))] \} \quad (3.13)$$

Note that  $(TV)(\cdot)$  is itself a value function defined over  $\mathbf{B}$ , so  $T$  is a mapping that transforms the value function  $V$  into another value function  $TV$ . Using this notation, (3.11) can be rewritten as

$$TV^*(b) = V^*(b).$$

The dynamic programming mapping has two important properties: it is monotonic and it is a contraction under the supremum metric. Therefore, the repeated application of the operator  $T$  to a value function  $V$  converges to the optimal value function (e.g., [26]). This result is the basis for the value iteration algorithm, which consists of evaluating a sequence of value functions  $V^n = TV^{n-1}$  until

$$\sup_{b \in \mathbf{B}} |V^*(b) - V^n(b)| < \varepsilon$$

where  $\varepsilon$  is an arbitrarily small error factor known as the Bellman error. In order to solve the infinite horizon problem, the operator  $T$  is applied to the value function  $V^1$  until  $|V^*(b) - V^n(b)| < \varepsilon$ , which can be expressed as a function of  $|V^n(b) - V^{n-1}(b)|$  (this is a standard result, found in [9] or [58], for example).

One important problem must be solved before value iteration can be used to find the optimal solution of (3.11): there is an infinitely large number of beliefs. The continuity of the belief space could mean that there exists some  $n$  for which  $V^n$  cannot be represented finitely or that the value function updates cannot be computed in a finite amount of time. Section 3.3.3 looks at how these difficulties can be overcome.

### 3.3.3 Finite Representation of the Value Function

One important property of POMDPs is that the finite-horizon value function is piecewise linear and convex. This implies that the optimal value function for the infinite horizon problem can be approximated arbitrarily well by a piecewise linear convex function. To be

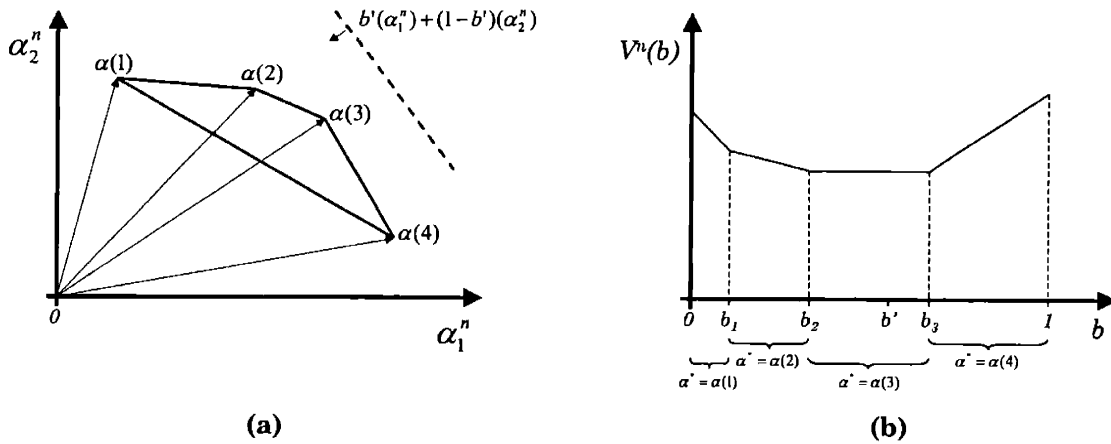


Figure 3-5: A set  $A^n$  of vectors and their corresponding value function

exact, the value function after  $n$  iterations of the value iteration algorithm can be written as

$$V^n(b) = \max_{\alpha \in A^n} (b \cdot \alpha)$$

where  $A^n$  is a finite set of  $|\mathbf{S}|$ -dimensional vectors. Figure 3-5a shows an example of the convex hull of a hypothetical set  $A^n = \{\alpha(1), \alpha(2), \alpha(3), \alpha(4)\}$  and 3-5b shows the corresponding value function  $V^n(b)$ .

The piecewise linearity and convexity of the value function is stated and proved in the form of a theorem below. This theorem is a special case of a more general result proved by Sondik [86]. It is important to go through the steps of the proof in this particular case because they give insight into the nature of the problem.

**Theorem 14** *The value function  $V^n(b)$ , obtained after  $n$  applications of the operator  $T$  to the value function  $V^1(b)$ , is piecewise linear and convex. In particular, there exists a set  $A^n$  of  $|\mathbf{S}|$ -dimensional vectors such that*

$$V^n(b) = \max_{\alpha \in A^n} (b \cdot \alpha).$$

**Proof.** The proof is by induction, and therefore consists of the verification of the base case and of the induction step.

(1) Base case:  $V^1(b) = \max_{\alpha \in A^1} (b \cdot \alpha)$ .

(2) Induction step: If

$$\exists A^{n-1} \text{ such that } V^{n-1}(b) = \max_{\alpha \in A^{n-1}} (b \cdot \alpha)$$

then

$$\exists A^n \text{ such that } V^n(b) = \max_{\alpha \in A^n} (b \cdot \alpha).$$

The statement above can be proved by verifying that the elements  $\alpha \in A^n$  are of the form

$$\alpha = p^A(X) + \gamma \Gamma^A(\alpha_i) + \gamma \Gamma^R(\alpha_j) \quad (3.14)$$

where  $\alpha_i \in A^{n-1}$ ,  $\alpha_j \in A^{n-1}$  and  $\Gamma^A$  and  $\Gamma^R$  are diagonal matrices defined by

$$\Gamma^A(X) = \begin{bmatrix} p_1^A(X) & 0 & \cdots & 0 \\ 0 & p_2^A(X) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_{|S|}^A(X) \end{bmatrix}$$

and

$$\Gamma^R(X) = \begin{bmatrix} p_1^R(X) & 0 & \cdots & 0 \\ 0 & p_2^R(X) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_{|S|}^R(X) \end{bmatrix}.$$

It then follows that

$$V^n(b) = \max_{\alpha \in A^n} (b \cdot \alpha)$$

which can be verified to be true by defining

$$A^n := \bigcup_{X \in \mathbf{X}} \left\{ \bigcup_{\alpha^i \in A^{n-1}} \left[ \bigcup_{\alpha^j \in A^{n-1}} (p^A(X) + \gamma \Gamma^A(X) \cdot \alpha^i + \gamma \Gamma^R(X) \cdot \alpha^j) \right] \right\}.$$

The details of the proof are in Appendix 3.8.1. ■

Theorem 14 provides the missing link to ensure that the value iteration algorithm is certain to arrive at the optimal solution for the finite-horizon problem and an arbitrarily good approximation for the infinite horizon problem in a finite amount of time. Unfortunately, this finite amount of time can be very large and often impractical. The essential difficulty of solving POMDPs lies in the efficient computation of the sets  $A^n$  since the size of these sets grows very fast with each step of the value iteration algorithm. More precisely, in each iteration of the algorithm there is one new  $\alpha$  vector for each possible product recommendation  $X$  for each permutation of size 2 of the vectors in  $A^{n-1}$ . This statement is formalized in the corollary below.

**Corollary 15**  $|A^n| = |\mathbf{X}|^{2^n - 1}$

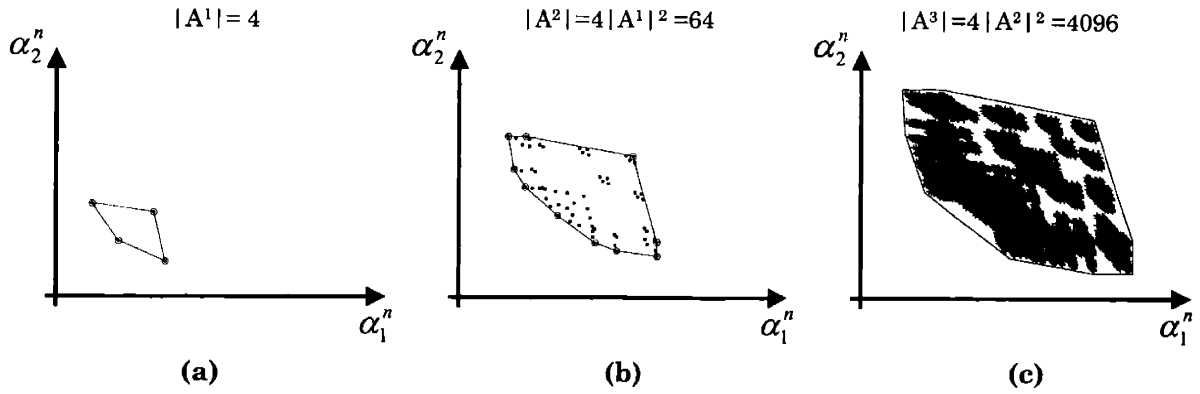


Figure 3-6: Generating sets  $A^n$  from  $A^{n-1}$

**Proof.** First, note that  $|A^1| = |\mathbf{X}|$ , which can be verified by recalling that

$$A^1 = \left\{ \left[ \begin{array}{c} p_1^A(X_1) \\ p_2^A(X_1) \\ \vdots \\ p_{|S|}^A(X_1) \end{array} \right], \dots, \left[ \begin{array}{c} p_1^A(X_{|\mathbf{X}|}) \\ p_2^A(X_{|\mathbf{X}|}) \\ \vdots \\ p_{|S|}^A(X_{|\mathbf{X}|}) \end{array} \right] \right\}.$$

Next, consider the set  $A^n$  single step of value iteration

$$A^n = \bigcup_{X \in \mathbf{X}} \left\{ \bigcup_{\alpha^i \in A^{n-1}} \left[ \bigcup_{\alpha^j \in A^{n-1}} (p^A(X) + \gamma \Gamma^A(X) \cdot \alpha^i + \gamma \Gamma^R(X) \cdot \alpha^j) \right] \right\}.$$

The size of  $|A^n|$  is:

$$|A^n| = \left| \bigcup_{X \in \mathbf{X}} \left\{ \bigcup_{\alpha^i \in A^{n-1}} \left[ \bigcup_{\alpha^j \in A^{n-1}} (p^A(X) + \gamma \Gamma^A(X) \cdot \alpha^i + \gamma \Gamma^R(X) \cdot \alpha^j) \right] \right\} \right| = |\mathbf{X}| |A^{n-1}|^2.$$

Solving the recursion  $|A^n| = |\mathbf{X}| |A^{n-1}|^2$  subject to the initial condition  $|A^1| = |\mathbf{X}|$  we conclude that

$$|A^n| = |\mathbf{X}|^{2^n - 1}.$$

■

The way in which the sets  $A^n$  generated make it impossible to determine the optimal value function in closed-form, and the rate at which they grow often make numerical solutions impractical. Figure 3-6 show the convex hull of a sequence of sets  $A^n$  for increasingly large  $n$ . The next two sections describe solution approaches that can be used in absence of a closed-form solution. Section 3.4 reviews methods that can be used to make the generation of the sets  $A^n$  more efficient in order to find the optimal solution numerically. Section 3.5 describes

suboptimal control policies that can be easily computed and implemented efficiently.

### 3.4 Exact Solution Methods

POMDPs were first introduced by Drake [29] in the context of telecommunications. Sondik ([86],[85], and [87]) made significant theoretical contributions to the understanding of the model. Applications in robot navigation resulted in a recent increase of interest in POMDPs in the artificial intelligence community (e.g., [48]) which led to very important contributions in computational methods. As pointed out in Section 3.3.3, the main difficulty in solving POMDPs is the increasing size of the sets  $A^n$  of vectors  $\alpha$ . Hence, the most important developments in exact solution method have been either (i) making the sets  $A^n$  as small as possible so as to minimize the size of  $A^{n+1}$  and (ii) finding ways of quickly generating the sets  $A^n$  from sets  $A^{n-1}$ .

Monahan [70] formulated a linear program that finds the smallest possible set  $A^n$  that contains all the vectors  $\alpha$  used in the optimal solution. Eagle [30] presents a slight modification of Monahan's algorithm. Littman's [58] Witness algorithm is one of the most important recent contributions to the efficient generation of  $A^n$  from  $A^{n-1}$ . The Witness algorithm efficiently builds a set  $Q^n$  such that  $Q^n \supseteq A^n$  and then uses Monahan's linear program or an equivalent procedure to reduce  $Q^n$  to  $A^n$ .

Figure 3-7 shows the results of numerical studies comparing the optimal policy  $\mu^*$ , defined in (3.12) and computed using the Witness algorithm, and the myopic policy  $\mu^M$ , which satisfies

$$\mu^M(b) = \arg \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) \right\}.$$

These results were obtained from simulations where companies started out with the same number of customers and gained no new customers over time. The graphs are typical of simulations done with varying numbers of customer segments and products. Graph (a) shows that the myopic policy is more likely to generate profits in the short-run, while the optimal policy does better in the long-run. The agent that acts optimally sacrifices sales in the beginning to learn about the customers to serve them better in the future. Graph (b) shows the cost of learning in terms of the customer base. The optimal policy leads to more customer defections due to low utility in the short-run, but the customers who stay receive consistently high utility in later periods and defect at much slower rates than customers being served by a myopic agent.

Exact solution methods, when feasible, are the best way to find the optimal solution. Littman et al. [59] describe methods to solve POMDPs with up to nearly one hundred states. However, for computational reasons, it is not always feasible to find the optimal solution. Furthermore, closed-form solutions are necessary to gain insight into how managerially-controllable variables can improve profit. The next section looks at easily computable policies that are better than the myopic solution and can be implemented efficiently.



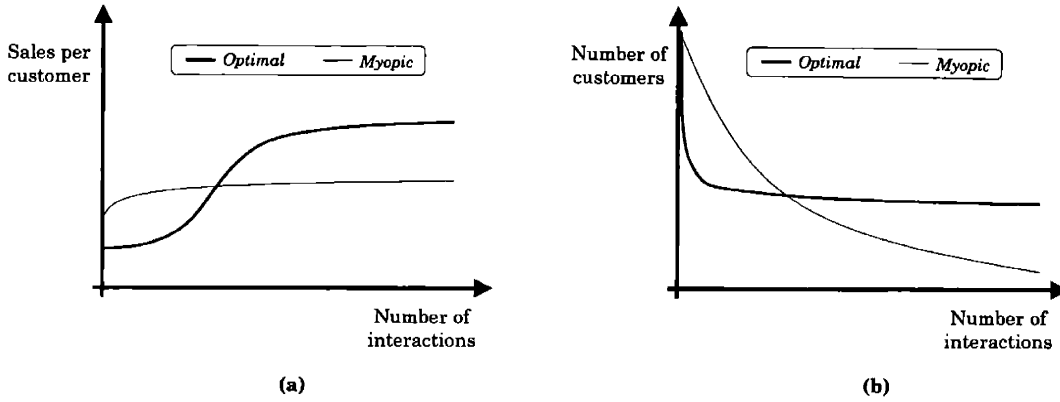


Figure 3-7: Comparison of optimal and myopic customization policies

### 3.5 Derivation of Control Policies

The optimal policy  $\mu^*(b)$ , which satisfies (3.12), cannot always be computed in a reasonable amount of time, as explained in Section 3.4. In the absence of the optimal policy, it would be desirable to have policies with the following characteristics:

- Convergence to the optimal policy.
- Lower the probability that the customer will leave after each interaction due to a bad recommendation.
- Better revenues than the myopic policy.
- Learn faster than the myopic policy.
- Ease of computation (or, better yet, closed-form solution).

An ideal control policy should have as many of these characteristics as possible. This section shows how control policies can be derived by finding approximations to the optimal value function.

#### 3.5.1 Value Function Approximations

Approximate value functions are useful tools in the generation of control policies when the optimal value function is not available. If  $V^*(b)$  can be approximated by  $\tilde{V}(b)$ , the corresponding policy  $\mu(b)$  satisfies

$$\tilde{\mu} = \arg \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \sum_{\theta \in \Theta} P(\theta|b, X) \tilde{V}(\varphi(b, \theta, X)) \right\}. \quad (3.15)$$

There are several ways of approximating  $\tilde{V}(b)$ . Bertsekas [9] discusses traditional approaches, and Bertsekas and Tsitsiklis [11] suggest recently developed methods based on neural networks and simulations. This section gives an example of two approximation methods that can be used to generate policies. Both methods were chosen because their control functions can be stated in simple terms, and are therefore better suited for insights into the nature of optimal customization policies in managerial settings.

### n-step Lookahead

The n-step Lookahead policy consists of approximating the optimal value function  $V^*(b)$  with the value function  $V^n(b)$ , obtained after  $n$  applications of the dynamic programming mapping (3.13). The  $n$ -step lookahead policy

$$\mu^{nLA} : \mathbf{B} \rightarrow \mathbf{X}$$

can be obtained by substituting  $V^n(b)$  for  $\tilde{V}(b)$  in equation (3.12):

$$\mu^{nLA} = \arg \max_{X \in \mathbf{X}} \{ \tilde{R}(b, X) + \gamma [P(\theta^A | b, X) V^n(\varphi(b, \theta^A, X)) + P(\theta^R | b, X) V^n(\varphi(b, \theta^R, X))] \}$$

One important special case of the n-step lookahead policy is the 1-step lookahead policy  $\mu^{1LA}$ , defined by

$$\mu^{1LA} = \arg \max_{X \in \mathbf{X}} \{ \tilde{R}(b, X) + \gamma [P(\theta^A | b, X) [\varphi(b, \theta^A, X) \cdot \tilde{R}^*(\varphi(b, \theta^A, X))] + P(\theta^R | b, X) [\varphi(b, \theta, X) \cdot \tilde{R}^*(\varphi(b, \theta^A, X))] ] \}$$

where

$$\tilde{R}^*(b) = \max_{X \in \mathbf{X}} [\tilde{R}(b, X)].$$

The theorem below shows how to estimate the difference between  $V^*(b)$  and  $V^n(b)$ .

**Theorem 16** *If  $V^k(b)$  is the value function obtained with the  $k$ -step lookahead policy and  $V^*(b)$  is the optimal value function, then*

$$\sup_{b \in \mathbf{B}} |V^k(b) - V^*(b)| \leq \frac{2\varepsilon\gamma^k}{1-\gamma}$$

where

$$\varepsilon = \sup_{b \in \mathbf{B}} |V^k(b) - V^{k-1}(b)|.$$

**Proof.** This is a standard result in infinite-horizon dynamic programming, and proofs can be found in textbooks such as [9]. ■

## Perfect Information

The policy derived from the perfect information value function approximation is given by

$$\mu^{PI} = \arg \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \sum_{\theta \in \Theta} P(\theta|b, X) V^{PI}(\varphi(b, \theta, X)) \right\}. \quad (3.16)$$

The perfect information stochastic shortest path problem is very easy to solve, as pointed out in § 3.2.3. The solution is stated and proved below.

**Proposition 17** *If  $V^{*FO}(S_i)$  denotes the optimal value function of the fully observable problem when the customer is of segment  $S_i$ , then*

$$V^{*FO}(S_i) = \max_{X \in \mathbf{X}} \frac{p_i^A(X_i)}{1 - \gamma(1 - p_i^L(X_i))}$$

and

$$V^{PI}(b) = b \cdot \max_{X \in \mathbf{X}} \frac{p_i^A(X_i)}{1 - \gamma(1 - p_i^L(X_i))}. \quad (3.17)$$

**Proof.** See Appendix 3.8.2. ■

The value function (3.17) can then be substituted into (3.16) to generate a control policy. Lovejoy [63] invokes a theorem from Van Hee [94] and notes that the bound can be tightened by applying the operator  $T$  (3.13):

$$TV^{PI}(b) \leq V^{PI}(b) \forall b.$$

Since the operator  $T$  is a contraction mapping, its repeated application generates an increasingly better sequence of upper bounds until it converges to a fixed point solution:

$$\begin{aligned} T^2 V^{PI}(b) &\leq TV^{PI}(b) \forall b, \\ &\vdots \\ T^n V^{PI}(b) &\leq T^{n-1} V^{PI}(b) \forall b \\ &\vdots \\ V^{PI*}(b) &= TV^{PI*}(b). \end{aligned}$$

Solving the fixed point equation yields

$$V^{PI*}(b) = b \cdot \max_{X \in \mathbf{X}} \frac{p_i^A(X_i)}{1 - \gamma(1 - p_i^L(X_i))} \geq V^*(b) \quad (3.18)$$

## Limited Learning

One way to derive control policies is to assume that the agent will stop learning after a fixed number of interactions. This restriction in the agent's actions implies that the same product

will always be recommended after a given point. These approximations are very good in situations where most of the learning occurs in the early stages.

Proposition 18 derives an expression for the policy  $\mu^{LL}(b)$ , which maximizes profits when agents are not allowed to change their beliefs between interactions.

**Proposition 18 3.8.3** *If agents are constrained to recommend the same product in all interactions with customers, then the infinite-horizon value function is given by*

$$V^{LL}(b) = \max_{X \in \mathbf{X}} \left\{ \sum_{i=1}^{|\mathbf{S}|} b_i \cdot \left( \frac{p_i^A(X)}{1 - \gamma(1 - p_i^L(X))} \right) \right\}. \quad (3.19)$$

and the corresponding policy by

$$\mu^{LL}(b) = \arg \max_{X \in \mathbf{X}} \left\{ \sum_{i=1}^{|\mathbf{S}|} b_i \cdot \left( \frac{p_i^A(X)}{1 - \gamma(1 - p_i^L(X))} \right) \right\}. \quad (3.20)$$

**Proof.** See Appendix 3.8.3. ■

The policy described in (3.20) can be re-written as

$$\mu^{LL}(b) = \arg \max_{X \in \mathbf{X}} \{ b' \cdot \Lambda(X) \cdot p^A(X) \}$$

where  $\Lambda(X)$  is a diagonal matrix that can be interpreted as the inverse of the risk associated with each recommendation for each segment. There are many ways to choose the elements of  $\Lambda(X)$ . In the particular case of (3.20), the risk is fully captured by the probability of leaving. In a real world application, this would be an appropriate place to incorporate the knowledge of managers into the control problem. If all the diagonal elements of  $\Lambda(X)$  are the same,  $\mu^{LL}$  reduces to the myopic policy. The matrix  $\Lambda(X)$  scales the terms  $p_i^A(X)$  by a risk factor unique to each product/segment combination. Therefore, under  $\mu^{LL}$  products that are very good (or very safe) for only a few segments are more likely to be chosen than products that are average for all segments. The opposite would be true with the myopic policy. Consequently, agents acting according to  $\mu^{LL}$  would observe more extreme reactions and have the potential to learn faster.

The policy described in (3.20) can be improved upon by allowing the agent to learn between interactions. In other words, the agent acts according to (3.20) but would update the beliefs after each interaction. This idea can be generalized to a class of policies where during each period the agent solves a  $n$ -period problem and chooses the action that is

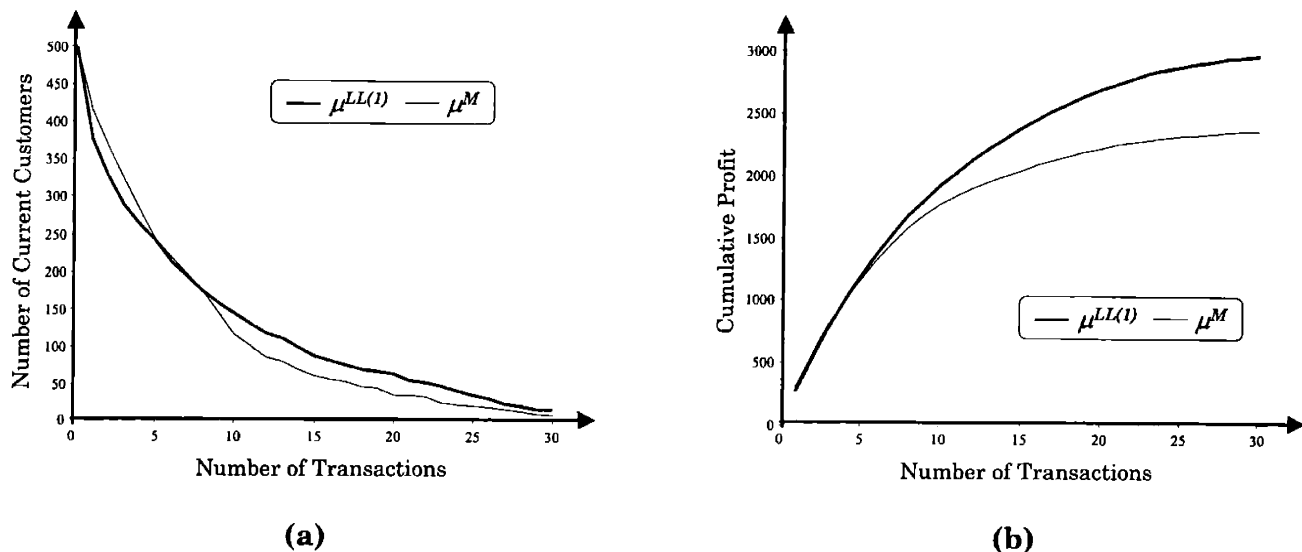


Figure 3-8: Comparing the policies  $\mu^{LL(1)}$  (limited learning) and  $\mu^M$  (myopic).

optimal for the first period, in a rolling-horizon manner. More precisely,

$$\begin{aligned} \mu^{LL(n)} &= \arg \max_{X \in \mathbf{X}} \left\{ E \left( \sum_{t=0}^n \gamma^t r_t \right) \right\} \\ &\quad s.t. \\ r_n(b) &= V^{LL}(b) \end{aligned}$$

In the specific case where  $n = 1$  the corresponding policy is given by

$$\mu^{LL(1)} = \arg \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \sum_{\theta \in \Theta} P(\theta|b, X) V^{LL}(\varphi(b, \theta, X)) \right\}. \quad (3.21)$$

Figure 3-8 shows the results of a numerical experiment designed to compare the performance of policies  $\mu^{LL(1)}$  and  $\mu^M$ . This experiment consisted of simulating purchase experiences for 500 customers. Figure 3-8a shows the customer base as a function of the number of interactions. The myopic policy takes less risks and performs better in the short-run, but agents using  $\mu^{LL(1)}$  learn faster and are able to retain a higher percentage of customers after 4 or so interactions. The net result is that policy  $\mu^{LL(1)}$  leads to a consistently higher customer base after 9 interactions. Figure 3-8b shows that the initial investment in learning by  $\mu^{LL(1)}$  pays off in the long-run in the form of higher cumulative profits.

### 3.5.2 Analyzing the Bounds

This section explains why using the policy  $\mu^{LL(n)}$  described in the § 3.5.1 consistently yields good results. The results of (3.19) and (3.18) can be combined to compute performance bounds as a function of the belief state:

$$\max_{X \in \mathbf{X}} \left\{ \sum_{i=1}^{|\mathbf{S}|} b_i \cdot \left( \frac{p_i^A(X)}{1 - \gamma(1 - p_i^L(X))} \right) \right\} \leq V^*(b) \leq \left\{ \sum_{i=1}^{|\mathbf{S}|} b_i \cdot \max_{X \in \mathbf{X}} \left( \frac{p_i^A(X)}{1 - \gamma(1 - p_i^L(X))} \right) \right\}.$$

Note, first of all, that the upper and lower bounds are very close to each other in the extreme points of the belief space. The worst-case scenario happens when the company knows very little about the customer. In this case, however, we would expect the upper bound to be very loose, since it is based on the assumption of perfect information. This suggests that using the lower bound value function as a basis for control may yield good results even if the upper and lower bounds are not very close. The next paragraph presents further arguments along these lines.

Figure 3-9 shows four possible configurations of products and customer segments. The first column refers to the situation when there are few customer segments and they are all very similar. Customization is of little value in this situation because one standardized product should suit all customer segments equally well. The cases described by the first row are equally uninteresting. If all products are very similar, the agent will learn very little by observing customers over time. However, profits will not be affected because the limited product variety would prevent the agent from catering to specific tastes in the first place. If the products that the agent can recommend are all very similar to each other, then the agent can pick any product and be confident that if that product is not optimal then it is at least very close to optimal. Cell (4) accurately captures the situation of online customization: different customer segments and different products. Since customers differ in their reactions to recommendations, early observations are very informative. This is precisely the situation when the policy performs well, because it observes “extreme” reactions from different segments and learns fast, moving to one of the regions in the belief space where  $\mu^{LL(1)}$  is more likely to coincide with the optimal policy. Therefore, it is not surprising that  $\mu^{LL(1)}$  performs so well empirically.

We conclude this section summarizing how policy  $\mu^{LL(1)}$  measures up to the criteria described at the beginning of this section.

#### 1. Convergence to the optimal policy

As the belief state approaches the extreme points of the belief space, the upper and lower bounds are very close to each other. This implies that over time, as the company learns about the customer, the policy described in (3.21) will be the same as the optimal policy.

#### 2. Lower the probability that the customer will leave after each interaction due to a bad recommendation.

This can only be guaranteed probabilistically because the observation error could the-

	Few, Similar Customers	Many, different Customers
Few, Similar Products	<b>1</b>	<b>2</b>
Many, different Products	<b>3</b>	<b>4</b>

Figure 3-9: Possible “extreme” configurations of customer segments and products

oretically be large enough to cause the customer to behave suboptimally. The likelihood of this event happening decreases as the utility of the recommendation increases, and the utility of the recommendations will increase over time.

**3. Better revenues than the myopic policy.**

When the company already knows the customer well,  $\mu^{LL(1)}$ , the myopic policy, and the optimal policy perform equally well. In situations where the agent needs to learn,  $\mu^{LL(1)}$  loses to the myopic policy in terms of short-run revenue, wins in terms of expected revenues over time.

**4. Learn faster than the myopic policy.**

The myopic policy is more likely to suggest products that have similar utilities across all customer segments when the agent is not sure about the customer’s true segment. This situation is likely to happen at the beginning of the relationship, and suggesting such products leads to little, if any, learning.

**5. Ease of computation (or, better yet, closed-form solution).**

The value function  $V^{LL}(b)$  can be described in closed-form. Therefore, policies  $\mu^{LL(n)}$  are very easy to compute and implement for small  $n$ .

## 3.6 Discussion

### 3.6.1 Solving the Agent's Dilemma

The main question of this paper is how companies should customize their products and services in settings such as the Internet. The analysis unambiguously shows that the predominant paradigm of maximizing the expected utility of the current transaction is not the best strategy. Optimal policies must take into account future profits and balance the value of learning with the risk of losing the customer. This paper describes two ways of obtaining good recommendation policies: (i) applying POMDP algorithms to compute the exact optimal solution (§3.4), (ii) approximating the optimal value function (§3.5.1).

The bounds analyzed in §3.5.2 make explicit why agents acting according to a myopic policy perform so poorly. These agents fail to recommend products that can be highly rewarding for some segments because they fear the negative impact that these recommendations could have on other segments. This fear is legitimate – companies can lose sales and customers can defect – but it can be quantified through a matrix  $\Lambda(X)$  that captures the risk associated with each possible recommendation for each segment. Optimally-behaving agents make riskier suggestions than myopic agents, learn faster, and earn higher payoffs in the long-run.

### 3.6.2 Learning and Loyalty

The more quality (utility) a company provides the customer, the more likely the customer is to come back. The better the company knows the customer, the more quality it can provide. Yet, learning about the customer can mean providing disutility (less quality) in the short-run. The interaction policy described in this paper reconciles these two apparently contradictory statements. The model suggests that the concept of loyalty must be expanded to incorporate learning. The extent to which repeated purchases lead to high value and loyalty depends on the company's ability to learn about their customers as they interact.

The benefits of loyalty to profitability have been abundantly discussed in the OM literature. Reichheld [78] gives the following examples of how loyalty leads to profits:

- In an industrial laundry service, loyal customers were more profitable because drivers knew shortcuts that made delivery routes shorter and led to cost reductions.
- In auto-service, if the company knows a customer's car, it has a better idea of the type of problem the car will have and will be able to identify and fix the problem more efficiently.
- A shop that sells to the same group of people year after year saves on inventory costs because it can stock only the items its customers like.



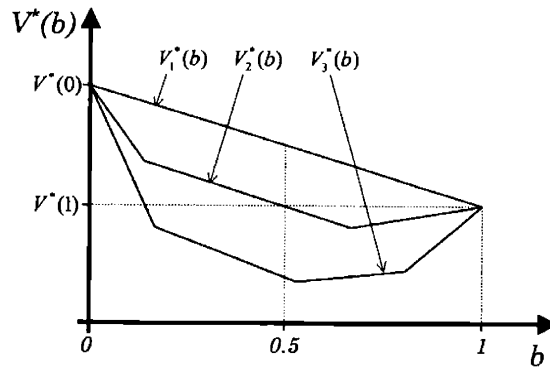


Figure 3-10: The value of customers as a function of how well the company knows them

These examples suggest an intimate relationship between learning and loyalty. In the laundry example, the fact that the customers have made a number of purchases from the same service provider has no direct effect on cost reduction. If a particular customer were to leave the company and its neighbor became a new customer, the cost savings would be the same in spite of the newness of the new customer. What seems to matter is knowledge, not length of tenure or number of previous purchases.

### 3.6.3 The Value of Knowing the Customer

The model presented in this paper shows how to quantify the value of learning about customers. Therefore, the optimal value function  $V^*(b)$  can be interpreted as the lifetime value of a customer as a function of how well the company knows that customer. The fact that  $V^*(b)$  is convex (as proved in §3.3.3) has interesting managerial implications.

Consider, for example, a company operating in an industry where customers of Segment 1 are half as profitable as customers in Segment 2. Which marketing campaign would be most profitable: (i) one that only brings in customers from Segment 1 or (ii) one that brings customers from both segments with equal probability? The answer is that it depends on the shape of the value function. Figure 3-10 shows three possible value functions. If  $V^*(b) = V_1^*(b)$ , then campaign (ii) is better; if  $V^*(b) = V_3^*(b)$ , then campaign (i) is better. Both campaigns are equally profitable if  $V^*(b) = V_2^*(b)$ . The intuition behind this result is that sometimes it can be costly to learn about one's own customers. In some situations the company is better off knowing for sure that they are providing excellent value to the least desirable segment. The lower the switching cost and the higher the variance in the utility observation error, the more companies should invest in targeted advertising.

The more a company knows the preferences of a particular customer, the more it should want to learn even more about that customer in order to transform knowledge into customized, highly valued services. Intuitively, one may think the information is most valuable

when the company doesn't know anything about the customer. The convexity of the value function implies the opposite. In qualitative terms, the change in the value of the customer as the company's level of knowledge changes from "very good" to "excellent" is greater than the change in value as knowledge goes from "average" to "good". This finding provides a theoretical justification for the empirical finding that the difference in loyalty between "very satisfied" and "satisfied" customers is much greater than the difference between "satisfied" and "slightly dissatisfied" customers [?]. If a company knows a customer well, that customer will receive consistently good recommendations and be very satisfied.

### **3.7 Directions for Future Research**

The results obtained in this paper drew on previous research in Management Science and Consumer Behavior. Future research in this topic can be pursued in both these directions. From the Management Science perspective, the issue of the best way to implement web-based customization policies remains open. The value function approximations derived on §3.5 lead to important insights, but the exact numerical methods of §3.4 generate better control policies. More work needs to be done to understand which approximation methods work best in situations where the problem is too large to compute the optimal policy. A second open question is how to recommend sets of products. In some situations agents are asked to recommend more than one product. One obvious way of addressing this issue is to consider each possible set as a single product and reduce the problem to the one solved in this paper. This method can be inefficient if the set of possible products is large, because of the exponential increase in the number of potential suggestions. Therefore, it is worthwhile to explore other ways to generate good policies.

From the Consumer Behavior perspective, there are two topics of interest. First, the development of efficient perception management devices to reduce the probability of defection. In some situations, it may be possible to have a "suggestion" to explore tastes as well as a "recommendation" to maximize utility. It has been shown that customers do not forgive bad recommendations, but could they forgive bad suggestions? Second, it is important to investigate how customers aggregate their experiences with agents over time. The model in this paper assumes that customer behavior is solely determined by the current product offering. In reality, the customers' perceptions of the agent's quality evolve over time. But how exactly does this evolution take place? On one hand, there is a stream of research which suggests that recency (the satisfaction derived in the last few interactions) is the most important determinant of current satisfaction. On the other hand, first impressions are important and it can be very difficult for a customer to change his or her initial perception of the agent's quality. These two views are in direct opposition, and it is not clear which is more important in web-based settings. Further research should investigate how customers' expectations change as a function of the quality of the advice they receive.

## 3.8 Appendix

### 3.8.1 Proof of Theorem 14

**Theorem:** The value function  $V^n(b)$ , obtained after  $n$  applications of the operator  $T$ , is piecewise linear and convex. In particular, there exists a set  $A^n$  of  $|\mathbf{S}|$ -dimensional vectors such that

$$V^n(b) = \max_{\alpha \in A^n} (b \cdot \alpha).$$

**Proof.** The proof is by induction, and therefore consists of the verification of a base case and the induction step.

(1) *Base case:* To prove that there exists a set of vectors  $A^1$  such that

$$V^1(b) = \max_{\alpha \in A^1} (b \cdot \alpha).$$

$V^1(b)$  is the value function with 1 step-to-go. The optimal policy with 1 step to go is to act myopically with respect to immediate payoffs, completely ignoring any learning that may occur. Specifically,

$$\begin{aligned} V^1(b) &= \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) \right\} \\ &= \max_{X \in \mathbf{X}} \left\{ \sum_{i=1}^{|\mathbf{S}|} b_i \cdot p_i^A(X) \right\} \end{aligned}$$

We can easily verify that  $V^1(b)$  is of the form

$$V^1(b) = \max_{\alpha \in A^1} (b \cdot \alpha).$$

by letting

$$A^1 = \left\{ \begin{array}{cc} p_1^A(X_1) & p_{|\mathbf{S}|}^A(X_1) \\ p_2^A(X_1) & p_{|\mathbf{S}|}^A(X_2) \\ \vdots & \vdots \\ p_{|\mathbf{S}|}^A(X_1) & p_{|\mathbf{S}|}^A(X_{|\mathbf{X}|}) \end{array} \right\}.$$

(2) *Induction step*

If

$$\exists A^{n-1} \text{ such that } V^{n-1}(b) = \max_{\alpha \in A^{n-1}} (b \cdot \alpha)$$

then

$$\exists A^n \text{ such that } V^n(b) = \max_{\alpha \in A^n} (b \cdot \alpha)$$

$V^n(b) = TV^{n-1}(b)$ , and, by equation (3.13), can be written as

$$V^n(b) = \max_{X \in \mathbf{X}} \{ \tilde{R}(b, X) + \gamma [P(\theta^A|b, X) V^{n-1}(\varphi(b, \theta^A, X)) \quad (3.22)$$

$$+ P(\theta^R|b, X) V^{n-1}(\varphi(b, \theta^R, X))] \}. \quad (3.23)$$

From the induction hypothesis, we can write  $V^{n-1}(\varphi(b, \theta^A, X))$  and  $V^{n-1}(\varphi(b, \theta^R, X))$  as

$$V^{n-1}(\varphi(b, \theta^A, X)) = \max_{\alpha \in A^{n-1}} (\varphi(b, \theta^A, X) \cdot \alpha) = \varphi(b, \theta^A, X) \cdot \alpha^{n-1}(b, \theta^A, X) \quad (3.24)$$

and

$$V^{n-1}(\varphi(b, \theta^R, X)) = \max_{\alpha \in A^{n-1}} (\varphi(b, \theta^R, X) \cdot \alpha) = \varphi(b, \theta^R, X) \cdot \alpha^{n-1}(b, \theta^R, X) \quad (3.25)$$

where

$$\alpha^{n-1}(b, \theta^R, X) = \arg \max_{\alpha \in A^{n-1}} (\varphi(b, \theta^R, X) \cdot \alpha)$$

and

$$\alpha^{n-1}(b, \theta^R, X) = \arg \max_{\alpha \in A^{n-1}} (\varphi(b, \theta^R, X) \cdot \alpha).$$

Substituting (3.24) and (3.25) into (3.22) yields

$$V^n(b) = \max_{X \in \mathbf{X}} \{ \tilde{R}(b, X) + \gamma \Pr(\theta^A|b, X) [\varphi(b, \theta^A, X) \cdot \alpha^{n-1}(b, \theta^A, X)] \quad (3.26)$$

$$+ \gamma \Pr(\theta^R|b, X) [\varphi(b, \theta^R, X) \cdot \alpha^{n-1}(b, \theta^R, X)] \}.$$

Recall the definition of  $\varphi(b, \theta, X)$  from equation (3.4) and note that  $\varphi(b, \theta^A, X)$  and  $\varphi(b, \theta^R, X)$  can be written as

$$\varphi(b, \theta^A, X) = \frac{b \cdot \Gamma^A(X)}{\Pr(\theta^A|b, X)}$$

and

$$\varphi(b, \theta^R, X) = \frac{b \cdot \Gamma^R(X)}{\Pr(\theta^R|b, X)}$$

where  $\Gamma^A(X)$  and  $\Gamma^R(X)$  are diagonal matrices defined by

$$\Gamma^A(X) = \begin{bmatrix} p_1^A(X) & 0 & \cdots & 0 \\ 0 & p_2^A(X) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_{|\mathbf{S}|}^A(X) \end{bmatrix} \quad (3.27)$$

and

$$\Gamma^R(X) = \begin{bmatrix} p_1^R(X) & 0 & \cdots & 0 \\ 0 & p_2^R(X) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_{|S|}^R(X) \end{bmatrix}. \quad (3.28)$$

If we substitute the definition of  $\tilde{R}(b, X)$  from equation (3.8) and the expressions (3.27) and (3.28) into equation (3.26) we note that  $V^n(b)$  simplifies to

$$V^n(b) = \max_{X \in \mathbf{X}} \{b \cdot p^A(X) + \gamma \cdot b \cdot \Gamma^A(X) \cdot \alpha^{n-1}(b, \theta^A, X) + \gamma \cdot b \cdot \Gamma^R(X) \cdot \alpha^{n-1}(b, \theta^R, X)\}$$

by cancelling out the terms  $\Pr(\theta^A|b, X)$  and  $\Pr(\theta^R|b, X)$ .

Factoring out the vector  $b$  yields

$$V^n(b) = \max_{X \in \mathbf{X}} \{b \cdot [p^A(X) + \gamma \Gamma^A(X) \cdot \alpha^{n-1}(b, \theta^A, X) + \gamma \Gamma^R(X) \cdot \alpha^{n-1}(b, \theta^R, X)]\}$$

Define the set  $A^n$  as

$$A^n = \bigcup_{X \in \mathbf{X}} \left\{ \bigcup_{\alpha^i \in \mathbf{A}^{n-1}} \left[ \bigcup_{\alpha^j \in \mathbf{A}^{n-1}} (p^A(X) + \gamma \Gamma^A(X) \cdot \alpha^i + \gamma \Gamma^R(X) \cdot \alpha^j) \right] \right\}$$

Finally, we can conclude that

$$V^n(b) = \max_{\alpha \in A^n} (b \cdot \alpha)$$

■

### 3.8.2 Proof of Proposition 17

**Proposition:** If  $V^{*FO}(S_i)$  denotes the optimal value function of the fully observable problem when the customer is of segment  $S_i$ , then

$$V^{*FO}(S_i) = \max_{X \in \mathbf{X}} \frac{p_i^A(X_i)}{1 - \gamma(1 - p_i^L(X_i))}$$

and

$$V^{PI}(b) = \sum_{i=1}^{|\mathbf{S}|} b_i \cdot \max_{X \in \mathbf{X}} \frac{p_i^A(X_i)}{1 - \gamma(1 - p_i^L(X_i))}. \quad (3.29)$$

**Proof.** The steady-state equations for the fully observable problems described in Figure 3-3 are given by a system of equations

$$V^{FO}(S_i, X) = p^A(X) + \gamma (p_i^A(X) + p_i^R(X)) V^{FO}(S_i) + p_i^L(X) V^{FO}(L). \quad (3.30)$$

There is one such equation for each segment  $S_i$ , and these equations are not connected since there are no arcs connecting the different segments. Using the fact that  $V^{FO}(L) = 0$ , we can rearrange the terms in (3.30) and simplify as follows:

$$V^{FO}(S_i, X) = \frac{p^A(X)}{1 - \gamma(p_i^A(X) + p_i^R(X))}.$$

The optimality condition can be stated

$$\mu^{FO*} = \arg \max_{X \in \mathbf{X}} \frac{p^A(X)}{1 - \gamma(p_i^A(X) + p_i^R(X))}$$

yielding the value function

$$V^{PI}(b) = \sum_{i=1}^{|\mathbf{S}|} b_i \cdot \max_{X \in \mathbf{X}} \frac{p_i^A(X_i)}{1 - \gamma(1 - p_i^L(X_i))}.$$

■

### 3.8.3 Proof of Proposition 18

**Proposition:** If agents are constrained to recommend the same product in all interactions with customers, then the infinite-horizon value function is given by

$$V^{LL}(b) = \max_{X \in \mathbf{X}} \left\{ \sum_{i=1}^{|\mathbf{S}|} b_i \cdot \left( \frac{p_i^A(X)}{1 - \gamma(1 - p_i^L(X))} \right) \right\}. \quad (3.31)$$

and the corresponding policy by

$$\mu^{LL}(b) = \arg \max_{X \in \mathbf{X}} \left\{ \sum_{i=1}^{|\mathbf{S}|} b_i \cdot \left( \frac{p_i^A(X)}{1 - \gamma(1 - p_i^L(X))} \right) \right\}. \quad (3.32)$$

**Proof.** If agents are not allowed to learn, then the value function that maximizes future expected profits for the  $n$ -period problem is given by

$$\hat{V}^n(b) = \max_{\alpha \in \hat{A}^n} (b \cdot \alpha).$$

where the sequence of sets  $\{\hat{A}^1, \hat{A}^2, \dots, \hat{A}^n\}$  is defined by the recursion

$$\begin{aligned}\hat{A}^1 &= A^1 \\ \hat{A}^2 &= \bigcup_{X \in \mathbf{X}} \left( \bigcup_{\alpha^i \in \hat{A}^1} [p^A(X) + \Gamma(X) \cdot \alpha^i] \right); \\ &\vdots \\ \hat{A}^n &= \bigcup_{X \in \mathbf{X}} \left( \bigcup_{\alpha^i \in \hat{A}^n} [p^A(X) + \Gamma(X) \cdot \alpha^i] \right)\end{aligned}$$

where  $\Gamma(X) = \Gamma^A(X) + \Gamma^R(X)$ , i.e.,

$$\begin{aligned}\Gamma(X) &= \begin{bmatrix} p_1^A(X) + p_1^R(X) & 0 & \dots & 0 \\ 0 & p_2^A(X) + p_2^R(X) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & p_{|S|}^A(X) + p_{|S|}^R(X) \end{bmatrix} \\ &= \begin{bmatrix} (1 - p_1^L(X)) & 0 & \dots & 0 \\ 0 & (1 - p_2^L(X)) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & (1 - p_{|S|}^L(X)) \end{bmatrix}\end{aligned}$$

Since the sets  $\hat{A}^n$  are defined recursively, the value functions  $\hat{V}^n(b)$  can be explicitly written as maximizations over  $\alpha \in \hat{A}^1$  as follows:

$$\begin{aligned}\hat{V}^1(b) &= \max_{\alpha \in \hat{A}^1} (b \cdot \alpha) \\ &= \max_{X \in \mathbf{X}} (b \cdot p^A(X)) \\ \hat{V}^2(b) &= \max_{\alpha \in \hat{A}^2} (b \cdot \alpha) \\ &= \max_{X \in \mathbf{X}} (b \cdot [p^A(X) + \Gamma(X) \cdot p^A(X)]) \\ &\vdots \\ \hat{V}^n(b) &= \max_{\alpha \in \hat{A}^n} (b \cdot \alpha) \\ &= \max_{X \in \mathbf{X}} (b \cdot [p^A(X) + \Gamma(X) [p^A(X) + \Gamma(X) [p^A(X) + \dots] \dots] \cdot p^A(X)]) \\ &= \max_{X \in \mathbf{X}} (b \cdot p^A(X) + b \cdot \Gamma(X) \cdot p^A(X) + \dots + b \cdot [\Gamma(X)]^{n-1} \cdot p^A(X)) \\ &= \max_{X \in \mathbf{X}} b \left[ \mathbf{I} + \sum_{i=1}^{n-1} [\Gamma(X)]^i \right] \cdot p^A(X)\end{aligned}$$

The sequence of functions  $\hat{V}^n(b)$  converges:

$$\begin{aligned}
\lim_{n \rightarrow \infty} \left( \hat{V}^{n+1}(b) - \hat{V}^n(b) \right) &= \tag{3.33} \\
&= \lim_{n \rightarrow \infty} \left( \max_{X \in \mathbf{X}} b \left[ \mathbf{I} + \sum_{i=1}^{n-1} [\Gamma(X)]^i \right] \cdot p^A(X) - \max_{X \in \mathbf{X}} b \left[ \mathbf{I} + \sum_{i=1}^{n-2} [\Gamma(X)]^i \right] \cdot p^A(X) \right) \\
&= \lim_{n \rightarrow \infty} \left( \max_{X \in \mathbf{X}} [\Gamma(X)]^n \cdot p^A(X) \right) \\
&= 0
\end{aligned}$$

The last step follows from the fact that all the elements in  $\Gamma(X)$  are in  $(0, 1)$ , and therefore  $\lim_{n \rightarrow \infty} [\Gamma(X)]^n$  is a matrix of zeroes. Now consider the difference

$$D^n(b) = b \cdot \hat{\alpha} - V^n(b)$$

where

$$\hat{\alpha} = \arg \max_{X \in \mathbf{X}} \left( b \cdot [\mathbf{I} - \Gamma(X)]^{-1} \cdot [p^A(X)] \right).$$

Substituting the full expression for  $V^n(b)$  yields

$$D^n(b) = \max_{X \in \mathbf{X}} \left( b \cdot [\mathbf{I} - \Gamma(X)]^{-1} \cdot [p^A(X)] \right) - \left( \max_{X \in \mathbf{X}} b \cdot \left[ \mathbf{I} + \sum_{i=1}^{n-1} [\Gamma(X)]^i \right] \cdot p^A(X) \right)$$

Since all the elements of  $\Gamma(X)$  are in  $(0, 1)$ , the expression  $[\mathbf{I} - \Gamma(X)]^{-1}$  is well-defined and can be expanded to

$$[\mathbf{I} - \Gamma(X)]^{-1} = \mathbf{I} + \sum_{i=1}^{\infty} [\Gamma(X)]^i.$$

Substituting the expansion above into the expression for  $D^n$  yields

$$D^n(b) = \max_{X \in \mathbf{X}} \left( b \cdot \left[ \sum_{i=n}^{\infty} [\Gamma(X)]^i \right] \cdot p^A(X) \right).$$

It then follows that

$$\lim_{n \rightarrow \infty} D^n(b) = 0, \forall b \tag{3.34}$$

The result from (3.33) ensures that the sequence  $V^n(b)$  converges to a limit  $V^*(b)$ . Equation (3.34) asserts that  $V^n(b)$  also converges to  $b \cdot \hat{\alpha}$ . By Theorem 7, the limit is unique, so  $\hat{\alpha}$  must be the policy that maximizes the infinite horizon value function. ■



# Chapter 4

## The Costs and Benefits of Information Acquisition

### 4.1 Introduction

Information is crucial to firms that offer customized services. The ability to provide services of high utility to customers depends on knowledge of the customers' preferences and the ability to customize the service offerings so as to meet those preferences. Firms can learn about their customers' preferences by asking them questions and by observing their behavior. Asking questions is one of the most effective ways to learn. Questions can be asked in many different ways, as can be seen in the extensive literature dealing with the issue of questionnaire and survey design (e.g., [98], [92]). Learning from observing interactions is significantly less obtrusive, but also entails costs.

In situations where firms and customers interact repeatedly over time firms can use the knowledge obtained by observing past interactions in order to server their customers better. However, learning from interactions may involve experimenting with service design and customization policies that can lead to unsuccessful interactions that can result in customer defections. This situation suggests that asking questions can be highly desirable. If the firm's objective is to sell as much as possible (or to give the customer as much utility as possible) then is it better to learn from observing interactions or by asking questions? The answer is that it depends on how much the firm is willing to pay the customer to answer questions. This is precisely the issue addressed in this chapter.

Asking questions can be expensive. This cost is not always explicit, as there are many reasons why asking questions may not be desirable. Unlike learning from observations, asking questions is obtrusive and makes it explicit to the customer that the firm is actively performing an activity that is directly related to service delivery. Customers go to agents precisely because they have search costs. Consequently, they must also have costs in wasting time teaching agents, whether it's by answering questions or by doing something else. Customers interacting with web-based interfaces can be particularly impatient when it comes to answering questions. The development of questionnaires and data collection can be very

expensive in some situations. Some service interfaces (e.g., wireless devices) do not allow for the implementation of questionnaires, and it may be necessary to incur the expense of calling individual customers to collect data.

Finding the optimal balance between asking questions and providing service is an important decision faced by firms that deliver services through automated interfaces. Firms always have a prior belief about their customers' preferences, but is that prior enough to start providing service or does the firm need more information about the customer? The agent needs to ask less questions upfront if it takes into account the learning that will occur from observing future interactions. This means the customer has to wait less to get service. On the other hand, in situation where customers are particularly sensitive to receiving bad service the firm can ask questions before delivering the service to minimize the probability of an unsuccessful service encounter.

This chapter is organized as follows. §4.2 provides a review of the relevant literature. §4.3 shows how to calculate the value of asking a question. This result help managers estimate how much they should be willing to invest in order to obtain more knowledge about their customers. In §4.4, the cost of asking questions is captured by the opportunity cost of not providing service. This section analyzes the problem of whether firms should use an encounter with a customer to provide service or to ask questions. §4.5 addresses the situation where the firm has a limited amount of resources that must be allocated to either learning about current customers or acquiring new customers. In this case, the cost of asking questions is captured by a reduction the flow of incoming customers. Finally, §4.6 offers some concluding remarks and managerial insights that follow from the analysis of the preceding sections.

## 4.2 Literature Review

The principle of the learning curve has been prominent in the Operations Management literature for decades. The general approach used states that the unit cost of production decreases (or productivity increases, as in [1]) as the number of units produced increases. This function exhibits diminishing marginal returns, as the cost converges to the lowest possible cost. The earliest applications were in airline manufacturing (e.g., [5] and [7]), but the concept has also been applied in a number of other settings (see [3] for a review containing more examples). The learning curve's simplicity and theoretical appeal has resulted in applications in a number of different settings, where the model usually displayed great predictive ability.

One important development in the application of learning curves to managerial situations is the generalization of the assumption of early models that the choice of learning curve was considered to be beyond the managers' control and taken to be a fixed input into decision-making models. Fine [34] shows how managers can effectively choose their learning curve by controlling the quality level at which they choose to operate their production systems. His model, known as the quality-based learning curve, explains that operating at higher quality levels leads to an increase in the rate of learning, and hence to faster achievements

of higher profitability levels. In [35] the quality-based learning curve is generalized to the case of a imperfect production process, where the system sometimes produces defective items. Whereas the level of quality was previously treated as a managerially-controlled variable, this model considers it to be the stochastic output of the inspection policy. This generalization leads to the conclusion that the learning benefits of quality control and inspection must be considered so as to prevent underinvestment in quality improvement activities. Marcellus and Dada [65] solve the problem of finding optimal strategies for investing in learning about an imperfect production process. They find that the optimal policy is to invest in learning until the probability of producing a defective product is sufficiently low. Dada and Marcellus [25] generalize their previous study ([65]) by applying in Fine's [35] operational context of a workstation. In this case, the decision maker must not only consider whether or not to perform maintenance (as in [35]) but also how much to invest in maintenance to learn about the possible reasons for defects to reduce the necessity of performing maintenance in the future (as in [65]). Their analysis concludes that the optimal policy is of the control-limit type where the manager does not choose to learn if the probability of failure is low enough.

The rate of learning depends on the rate of production, and since the rate of production varies over time learning per unit time is also variable. This limitation of the traditional learning curve model was acknowledged in the early days by Asher [5] and mathematically formalized by Alchian [2]. Variable learning rates were a relatively simple theoretical addition to the learning curve model, but they raise estimation problems which were only addressed several years later by Gulledge, Tarimcilar, and Womer [41].

Rosen [79] made the distinction between learning by doing and learning by other means, a concept subsequently generalized to the concept of "knowledge creation" (e.g., [18]). This is an important distinction, as the term "experience curve" is often assumed to mean "learning curve" in spite of the fact that learning can be acquired by means other than experience, such as investment in quality and knowledge transfer. Mody [69] analyzed the case where production resources could be allocated to knowledge discovery through "investment in engineering". Killingsworth [51] took a similar approach in his analysis of "investment in training".

Carillo and Gaimon [18] introduce a model that shows how process changes which lead to short-term losses can lead to increases in long term capacity and hence profitability. In their model, knowledge can be created in two ways: learning by doing and learning before doing. Acquired knowledge leads to process changes, which are assumed to lead to higher levels of profitability. Terwiesch and Bohn [89] extend this model to a dynamic optimization framework where the firm has limited resources that can either be used to increase volume and reap the benefits of selling in the present period or to learn so as to improve future yield. Thomke and Bell [90] note that knowledge creation can occur at any point throughout the production process. In their model, knowledge creation is operationalized through sequential experiments through which managers learn about technical and customer-based aspects of their product. They solve the optimization problem of the optimal number and timing of experiments, and find an EOQ-like result where the optimal number of tests is the square

root of the ratio of the avoidable costs and the cost of a test.

Dorroh, Gullidge, and Womer [28] present a model where managers have the option of making direct investments in learning. This assumption radically breaks away from previous models where investments in learning took the form of allocating constrained resources from production or engineering to knowledge creation. In this sense, this is the closest model to the one introduced in this chapter. However, there are some important differences. Their model is applied in a manufacturing context where a firm produces specialized units to contractual order. The learning curve in this context refers to cost reductions or increases in yield (per fixed capacity or time unit), both of which have a linear impact on the firm's profit function. Therefore, one of their main findings is the rate of investment declines as the production program matures. In this chapter, profit (or the value of the customer) is a convex function of knowledge, and the most interesting findings relate to changes in the rate of investment as knowledge increases.

### 4.3 The Value of Information

Consider a firm that provides a service that can be configured in many different ways to customers from different segments. Each customer belongs to a segment  $S_i \in \mathbf{S}$ . The firm's beliefs about the segment to which a given customer belongs are represented by a  $|\mathbf{S}|$ -dimensional vector  $b \in \mathbf{B}$ , with each component corresponding to the probability that the customer belongs to a given segment. The components of the belief vector have the properties that  $b_i \in [0, 1]$ ,  $\forall i$  and  $\sum_{i=1}^{|\mathbf{S}|} b_i = 1$ .

The value function represents the customer lifetime value as a function of how well the firm knows the customer. This function is convex, which means that the value of obtaining an additional piece of information about the customer increases the more the firm already knows the customer

Suppose that the current belief about a certain customer is  $\hat{b}$  and the firm has the option of asking the customer a question  $Y$ . Suppose that the question can only be answered in two ways,  $Y(1)$  and  $Y(2)$ . If the customer answers  $Y(1)$ , the belief will change to  $\hat{b}_1$ ; if he answers  $Y(2)$  the belief changes to  $\hat{b}_2$ . This situation is depicted in Figure 4-1. If the belief is  $\hat{b}_1$ , then the customer would be worth  $V(\hat{b}_1)$  to the firm. If it is  $\hat{b}_2$ , then the customer would be worth  $V(\hat{b}_2)$ . Therefore, the expected value of the customer after the question is answered is

$$V(Y|b) = \Pr(Y(1)|b) V(\hat{b}_1) + \Pr(Y(2)|b) V(\hat{b}_2).$$

If the firm incurs a cost  $c$  in order to ask this question, they should ask the question whenever

$$V(\hat{b}) + c < \Pr(Y(1)|\hat{b}) V(\hat{b}_1) + \Pr(Y(2)|\hat{b}) V(\hat{b}_2).$$

More generally, let  $\mathbf{Y} = \{Y_1, Y_2, \dots, Y_{|\mathbf{Y}|}\}$  be the set of all possible questions that can be

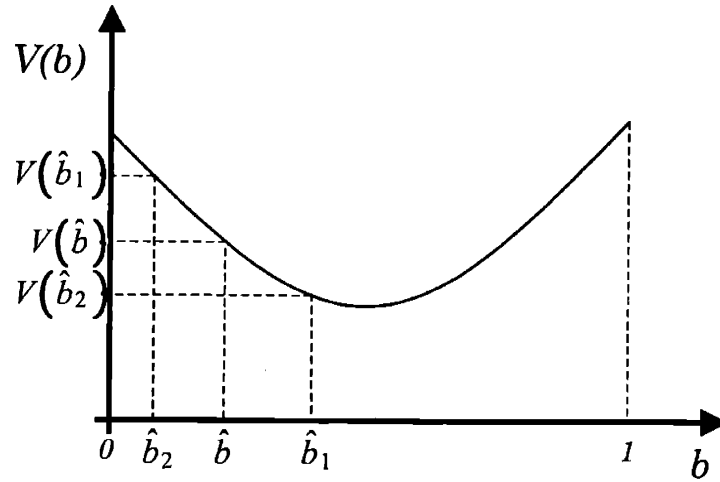


Figure 4-1: The value of asking a question

asked. Let  $Y_i^R = \{Y_i(1), Y_i(2), \dots, Y_i(|Y_i^R|)\}$  be the set of possible responses to the question  $Y_i$ , and let  $c(Y_i)$  be the cost of asking question  $Y_i$ . The value of asking question  $Y_i$  when the belief is  $\hat{b}$  is denoted  $V(Y_i|\hat{b})$  and it is given by

$$V(Y_i|\hat{b}) = \sum_{j=1}^{|Y_i^R|} \Pr(Y_i(j)|\hat{b}) V(\hat{b}_j) - V(\hat{b}). \quad (4.1)$$

If the current belief is  $\hat{b}$  then the firm should choose to ask a question whenever

$$\exists Y_i \in \mathbf{Y} \text{ s.t. } V(Y_i|\hat{b}) > V(\hat{b}) + c(Y_i). \quad (4.2)$$

The expression in (4.2) provides the necessary conditions to ensure that the firm asks a question whenever it is optimal to do so. One immediate corollary from this condition is that firms should always ask questions when there is no cost in asking. In other words, if the costs is the same then more information is always better than less information. The usefulness of this expression in real-life situations is limited in that the cost of asking questions is not always directly quantifiable. In some cases, the cost can be the opportunity cost of not making a product recommendation. In other cases, firms must decide whether to allocate their marketing resources to learning about their current customer or in acquiring new customers. The next two sections of this chapter address each of those two situations.

## 4.4 To Provide Service or to Ask Questions?

In this section we analyze the decision problem faced by firms that must decide whether to provide service or to ask questions. This is a problem constantly faced by firms that communicate with customers through permission email. Permission email (REF) is a system in which clients agree to receive personalized email messages with a pre-specified frequency. There are many different types of firms using the channel to communicate with customers. Permission email is common in industries such as news services, airlines, and a variety of retailers. Customized messages must be chosen carefully because customer defection is often only one mouse-click away. The content of the messages can take a variety of forms. In this paper, we consider the cases where the customized message can contain a question or a recommendation. If the firm asks a question there is no chance of making a profit in that interaction, but the chance that the customer will leave due to bad service is very small. Recommendations can yield profit, but a bad recommendation can be perceived as a sign of poor service and can therefore lead to defections.

Suppose that the firm has at its disposal a set  $\mathbf{Y} = \{Y_1, Y_2, \dots, Y_{|\mathbf{Y}|}\}$  of questions that it can ask and a set  $\mathbf{X} = \{X_1, X_2, \dots, X_{|\mathbf{X}|}\}$  of services it can provide. Customers behave according to a random utility model. A customer from segment  $i$  who is offered service  $X$ , will accept the offering with probability  $p_i^A(X)$ , reject it and come back for future interactions with probability  $p_i^R(X)$  and leave the firm with probability  $p_i^L(X)$ . Similarly, a customer from segment  $i$  who is asked question  $Y$ , will give answer  $Y(j)$  with probability  $p_i^{Y(j)}(Y)$  and leave the system with probability  $p_i^L(Y(j))$ . We assume that the firm earns a payoff of 1 when the customer accepts a recommendation and 0 otherwise. We also assume that firms never earn a payoff when they ask a questions, and that the probability of leaving due to asking a questions is lower than the probability of leaving when the customer is offered a product.

Let  $\mathbf{Z} = \mathbf{X} \cup \mathbf{Y}$  denote the set of all possible actions that the firm can take. Also, let  $\Theta$  be the set of all the possible observations that h firm can make after the customer reacts to the action. The set  $\Theta$  includes observing the actions of accepting, rejecting, or leaving (if the firm chooses an action from  $\mathbf{X}$ ) and possible answers to the questions in  $\mathbf{Y}$ . Furthermore, let  $b^t$  denote the belief vector after  $t$  interactions. The components of  $b^t$  are defined recursively according to the Bayesian rule

$$b_i^t = \Pr(S^* = S_i | b^{t-1}, Z^t, \theta^t) = \frac{1}{\Pr(\theta^t | b^{t-1}, Z^t)} [b_i^{t-1} \Pr(\theta^t | S_i, Z^t)] \quad (4.3)$$

where

$$\Pr(\theta^t | b^{t-1}, Z^t) = \sum_{i=1}^{|\mathbf{S}|} [b_i^{t-1} \Pr(\theta^t | S_i, Z^t)].$$

It is also useful to define an update function  $\varphi(b, \theta, Z)$ , which maps beliefs, actions, and

observations into new beliefs. This function is derived directly from (4.3) and is given by

$$\varphi(b, \theta, Z) = \frac{1}{\Pr(\theta|b, Z)} \cdot \begin{pmatrix} b_1 \cdot \Pr(\theta|S_1, Z) \\ b_2 \cdot \Pr(\theta|S_2, Z) \\ \dots \\ b_{|S|} \cdot \Pr(\theta|S_{|S|}, Z) \end{pmatrix}.$$

The expected revenue per period is a function of the belief and the action taken. It is denoted  $\tilde{R}(b, Z)$  and it is determined by the expressions

$$\tilde{R}(b, Z) = \sum_{i=1}^{|S|} b_i \cdot p_i^A(Z) \text{ if } Z \in \mathbf{X}$$

and

$$\tilde{R}(b, Z) = 0 \text{ if } Z \in \mathbf{Y}.$$

There are two ways to approach the problem of deciding whether to ask a question or to make a recommendation. In the first approach, firms assume that at each point in time they will always have the choice of asking any of the available questions. In the second approach, firms have the choice of either asking a question or providing service in the first period but are constrained to provide service in all the future interactions. The set of policies available in the second approach is a subset of the policies available in the first approach, implying that a solution that uses the first approach will always be dominant. However, the second approach has number of advantages. First, it is easier to compute. Second, it provides a better description of policies used and available to firms operating in the real world. Finally, when the policy can be implemented in a rolling-horizon basis, in which case the firm would always have the option of asking questions. We now turn to the analysis of each of these two cases.

#### 4.4.1 One-Stage Interactions

The decision process analyzed in this section is depicted in Figure 4-2. At each point in time, the firm has the option of asking any of  $|\mathbf{Y}|$  questions of recommending any of  $|\mathbf{X}|$  products.

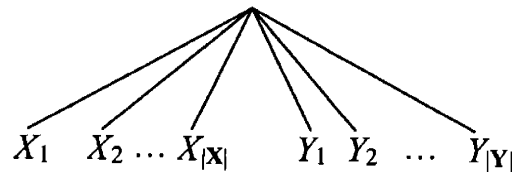


Figure 4-2: One-stage decision process

This decision problem can be modeled as a POMDP. The optimal policy is a mapping  $\mu$  defined by the expression

$$\mu^*(b) = \arg \max_{Z \in \mathcal{Z}} \left\{ E \sum_{t=0}^{\infty} \gamma^t r_t \right\}$$

where  $\gamma$  is a discount factor and  $r_t$  is the revenue earned during the  $t$ 'th interaction. The optimal value function must satisfy the Bellman Equation

$$V^*(b) = \max_{Z \in \mathcal{Z}} \left\{ \tilde{R}(b, Z) + \gamma \sum_{\theta \in \Theta} [P(\theta|b, Z) V^*(\varphi(b, \theta, Z))] \right\},$$

which implies that the optimal policy can be determined through the expression

$$\mu^*(b) = \arg \max_{Z \in \mathcal{Z}} \left\{ \tilde{R}(b, Z) + \gamma \sum_{\theta \in \Theta} [P(\theta|b, Z) V^*(\varphi(b, \theta, Z))] \right\}.$$

The optimal value function can be calculated exactly or approximated by a number of methods, many of which are described in Chapter 2.

#### 4.4.2 Two-Stage Interactions

The decision process analyzed in this section involves modeling the agent's decision problem as a two stage model. The two-stage model (not to be confused with the two-period model, where the customer and the firm only interact twice) consists of analyzing the relationship between customers and agents as having a period where the agent has the option of learning about the customer by asking questions and then the agent is constrained to provide service on all subsequent interactions. The decision trees corresponding to the decisions to be made during each of the stages is depicted in Figure 4-3. In the first stage, depicted in Figure 4-3(a) the firm can offer a service  $X_i \in \{X_1, X_2, \dots, X_{|X|}\}$  or ask a question  $Y_i \in \{Y_1, Y_2, \dots, Y_{|Y|}\}$ . Figure 4-3(b) shows the possible actions for all subsequent periods, i.e., always to provide service.

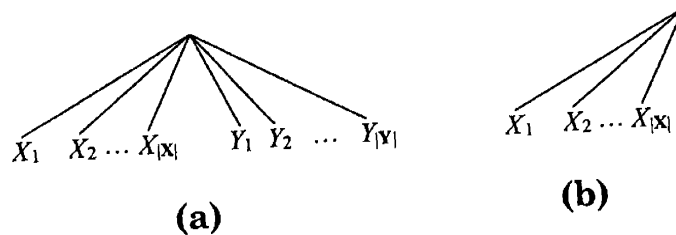


Figure 4-3: Decision trees for the two-stage decision process



The decision problem faced from the second period onwards is the solution of a POMDP with  $|\mathbf{X}|$  controls. This is exactly the problem that was solved in Chapter 3. The decision in the first stage can be found by acting greedily with respect to the value function obtained as the solution of this POMDP. two separate POMDPs. This result is important because it greatly reduces the computational burden of finding the optimal solution. The  $n$ 'th update of the value iteration algorithm involves the computation of only  $|\mathbf{X}|^{2^n-1}$ , as opposed to  $|\mathbf{X} + \mathbf{Y}|^{2^n-1}$  in the case of the 1-step formulation.

Let  $\mu_X^*(b)$  be the policy that fulfills the condition

$$\mu_X^*(b) = \arg \max_{X \in \mathbf{X}} \left\{ \tilde{R}(b, X) + \gamma \sum_{\theta \in \Theta} [P(\theta|b, X) V^*(\varphi(b, \theta, X))] \right\}$$

and let  $\mu_Y^*(b)$  be the solution of

$$\mu_Y^*(b) = \arg \max_{Y \in \mathbf{Y}} \left\{ \gamma \sum_{\theta \in \Theta} [P(\theta|b, Y) V^*(\varphi(b, \theta, Y))] \right\}.$$

These two equations can be combined with (4.1) to generate a rule to decide whether to provide service or to ask a question in the first period. The rule is to ask a question whenever

$$V(\mu_Y^*(b)|b) > \tilde{R}(b, \mu_X^*(b)) + \gamma \sum_{\theta \in \Theta} [P(\theta|b, \mu_X^*(b)) V^*(\varphi(b, \theta, \mu_X^*(b)))].$$

## 4.5 Learning vs. Customer Acquisition

The decision of whether to ask a question or provide a service involves a tradeoff between acquiring information and generating profits. This tradeoff can take a different form in situations where the firm can pay their customers in order to learn about them. This payment can occur through benefits such as frequent flyer miles, discounts, or coupons. In these situations, firms must be able to estimate the value of the information they wish to acquire so they can devise their information acquisition policies accordingly. The value of information is determined by the way in which the information will be used. Therefore, we must begin by describing the setting in which the firm operates.

This chapter considers the setting where firms and customers interact over time. In particular, the focus is on situations where firms customize the service so as to maximize the total expected profits in the long run. This means that there is a customization policy that maps beliefs about the customers' preferences into service configurations (or customized products, as the case may be). Customers can react to these recommendations in three different manners (see §4.4 for a more detailed description of the dynamics) and firms learn by observing the customers' reactions. As firms learn about customers they provide increasingly better recommendations. This means that the higher the level of knowledge about customers the lower the probability the customer will leave due to receiving a bad recommendation. The

option to purchase information instead of learning through observations can be beneficial in two ways. First, it will enable companies to earn levels of profit that would otherwise be only attained after interacting with the customer for a long period of time. Second, it reduces the probability that the customer will leave due to receiving a very bad recommendation.

### 4.5.1 Quantifying the value of learning

#### The Value of Customers as a function of Knowledge

In this section, we introduce a new variable, “knowledge” which represents how well the firm knows the customer regardless of the segment to which the customer belongs. We define the domain of this function to be  $[0,1]$  and assume that it has the following properties. First, the value of customers increases as a function of knowledge, i.e.,

$$\frac{dV(k)}{dk} > 0. \tag{4.4}$$

Second, we assume that the function  $V(k)$  is convex. We know from §4.3 that the lifetime value of customers is a convex function of the belief. As the belief state approaches the corners of the belief simplex, the belief vector approaches a vector of all zeros except for one of the elements, which has value 1. Based on this fact, it is reasonable to assume that the value of customers is a convex function of knowledge, and we write

$$\frac{d^2V(k)}{dk^2} > 0. \tag{4.5}$$

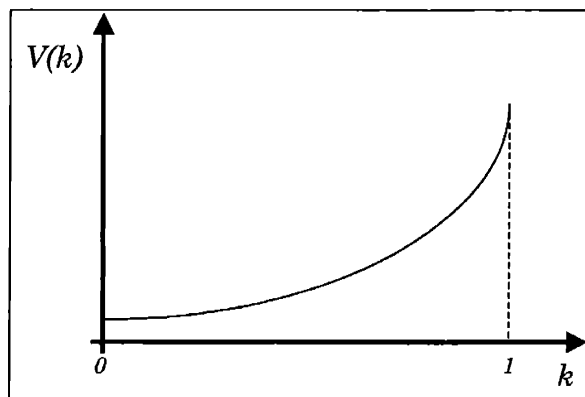


Figure 4-4: The value of customers as a function of knowledge

Finally, we note that the value of the customer is always finite. Even if the company knows the customer perfectly well the amount of profit it is able to extract from that customer is not infinite, since the customer can leave or payoffs can be discounted over time. This

property is expressed as

$$V(1) < \infty. \quad (4.6)$$

In the value function described above, knowledge can be thought of as an abstraction of the belief vector. If a customer reacts to a recommendation in an unexpected manner the belief vector can change significantly, causing the firm to believe that the customer is more likely a member of a different segment than was previously assumed. Regardless of the outcome, knowledge increases and so does the value of the customer.

Let  $C$  denote the customers currently in the system. The value of the customer base is

$$V = \sum_{c=1}^C V(k_c),$$

where  $k_c$  is the level of knowledge of the  $c$ 'th customer.

### The Learning Function

In order to model the value of learning, we introduce a function  $L(m_L; k)$  that maps the current level of knowledge ( $k$ ) and the resources allocated to learning ( $m_L$ ) into a new level of knowledge. More precisely,

$$L : K, M \rightarrow K.$$

The set  $K$  represents the possible states of knowledge, and corresponds to the set  $[0, 1)$ . As in random utility models, it is assumed that perfect knowledge, i.e.,  $k = 1$  is unattainable in practice.  $M$  are the possible levels of resource allocation to learning, and it takes values in the interval  $[0, \hat{m}]$  where  $\hat{m}$  is the total amount of resources available.

The first property of the learning function defined in this section is that if no resources are invested, the state of knowledge remains the same:

$$L(0; k) = k \quad (4.7)$$

for all  $k$ . This property is based on the assumption that customers do not change tastes over time, and can eventually be relaxed.

The second property of the learning function is that knowledge always increases if firms allocate resources to learning, i.e.,

$$\frac{\partial L(m_L; k)}{\partial m_L} > 0 \text{ if } m_L > 0. \quad (4.8)$$

It is more difficult to obtain knowledge about customers once the firm already knows something about them. When the firm does not know anything about the customers, simple questions can make a large difference in improving the level of knowledge. As firms learn about customers, they seek to refine their already good knowledge about customers. This means that questionnaires can become longer and it is more expensive to learn. The inability

to attain perfect knowledge is a feature of all random utility models, which assume that the reactions of customers can vary in each particular interaction. This property is captured by the expression

$$\lim_{m_L \rightarrow \infty} L(m_L; k) = 1 \tag{4.9}$$

and implies that

$$\lim_{m_L \rightarrow \infty} \left( \frac{\partial L(m_L; k)}{\partial m_L} \right) = 0.$$

Another important property that the learning function must satisfy is that the amount of learning that occurs must depend only on the initial level of knowledge and the total amount of resources invested, not on the order in which resources were invested. Suppose that the firm is willing to invest  $(m_{L_1} + m_{L_2})$  in order to learn about a customer currently at knowledge level  $k$ . If invests  $m_{L_1}$ , the knowledge will change to  $L(m_{L_1}; k)$ . If once the knowledge has reached this level an additional  $m_{L_2}$  is invested, then the knowledge will change to  $L(m_{L_2}; L(m_{L_1}; k))$ . If, on the other hand, the company invests all resources at once, the resulting knowledge level will be  $L(m_{L_1} + m_{L_2}; k)$ . This must be the same level of knowledge attained by making the investment sequentially. The analogous concept in information theory is the information obtained from two independent events is the sum of the amounts of information obtained from the single events (REF). This property is represented by the functional equation

$$L(m_{L_2}, (L(m_{L_1}; k))) = L(m_{L_1}, (L(m_{L_2}; k))) = L(m_{L_1} + m_{L_2}; k) \tag{4.10}$$

Figure 4-5 shows the function  $L(m_L; k)$  for different values of  $k$ .

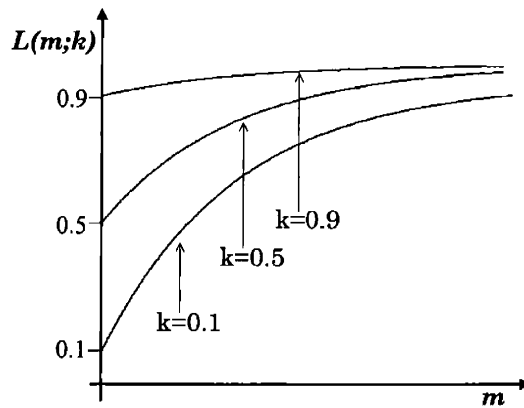


Figure 4-5: Learning functions for different levels of initial knowledge

The analysis that follows will make extensive use of the inverse learning function  $L^{-1}(k_1, k_2)$ . The inverse learning function is the amount of investment that is necessary to achieve knowledge  $k_2$  if the initial level of knowledge is  $k_1$ . This function is defined by the equivalence

relationship

$$L^{-1}(k_1, k_2) = m \iff L(m; k_1) = k_2. \quad (4.11)$$

It follows immediate from (4.11) that if  $L(m_1, k_1) = k_2$  and  $L(m_2, k_2) = k_3$  then

$$L(L^{-1}(k_1, k_2) + L^{-1}(k_2, k_3), k_1) = k_3. \quad (4.12)$$

The Lemma below proves a useful result about the inverse learning function that will be important in the analysis contained later in this chapter.

**Lemma 19** *If  $k_1 < k_2$  then for all  $k \in (k_1, k_2)$*

$$L^{-1}(k_1, k_2) = L^{-1}(k_1, k) + L^{-1}(k, k_2)$$

**Proof.** Since  $L$  is a learning function, it must satisfy (4.12). Therefore, we have

$$\begin{aligned} L(L^{-1}(k_1, k_2); k_1) &= L(L^{-1}(k, k_2); L(L^{-1}(k_1, k); k_1)) \\ &= L(L^{-1}(k_1, k) + L^{-1}(k, k_2); k_1) \quad \blacksquare \end{aligned}$$

## Customer Arrival Process

Suppose that customer arrive according to a Poisson process with rate  $\lambda(m_A)$ . We assume that the higher the amount of resources allocated to customer aquisition ( $m_{aA}$ ) the higher the customer arrival rate  $\lambda(m_A)$ . We also assume that after a certain amount of investment the firm cannot increase the arrival rate significantly, i.e., there are diminishing returns. These assumptions are based on models of customer response to advertising expenditure (e.g., [56], [57]). If It then follows that

$$\begin{aligned} \frac{\partial \lambda(m_A)}{\partial m_A} &> 0; \\ \lim_{m_a \rightarrow \infty} \frac{\partial \lambda(m_A)}{\partial m_A} &< 0. \end{aligned}$$

Little [57] points out that response to advertising can be concave or S-shaped, so any assumptions about the second derivative of  $\lambda(m_a)$  with respect to  $m_a$  will depend on the specific setting to which the model is being applied.

The probability distribution of the number of customer that arrive between interactions is given by

$$\Pr(N = n) = e^{-\lambda(m_A)} \frac{(\lambda(m_A))^n}{n!}, \quad (4.13)$$

and the expected value of this random variable is given by

$$E(N) = \lambda(m_A).$$

## 4.5.2 The Resource Allocation Problem

Let  $\hat{m}$  be the total available resources,  $m_A$  the resources allocated to customer acquisition and  $m_L$  the resources allocated to learning about customers. If we also assume that resources cannot be allocated for any other purpose it follows that

$$m_A + m_L = \hat{m}.$$

The optimization problem considered in this section is how to allocate  $\hat{m}$  units of resources to customer learning or customer acquisition. The analysis begins with the study of simplified cases which provide insight into the nature of the optimal allocation policies. The results from these analyses serve as building blocks that are combined to yield the optimal solution, presented in Theorem .

### One customer, constant acquisition rate

**Unlimited Resources** Suppose that the firm has one customer and its knowledge of that customer is  $k$ . Then the value of the customer base is  $V(k)$ . If an investment of  $m_L$  is made to learn about the customer, the value of the customer base will change to  $V(L(m_L; k))$ . Therefore, the revenue from the investment is  $[V(L(m_L; k)) - V(k)]$ , and the profit from making the investment is given by the function

$$\Pi(m_L; k) = V(L(m_L; k)) - V(k) - m_L. \quad (4.14)$$

Regardless of the initial level of knowledge, the profit always goes to minus infinity as  $m$  goes to infinity. This result is formalized below.

**Lemma 20**  $\lim_{m \rightarrow \infty} \Pi(m; k) = -\infty$  for all  $k$ .

**Proof.**  $\Pi(m; k) = V(L(m; k)) - V(k) - m$   
 $\lim_{m \rightarrow \infty} V(L(m; k)) = 1$  follows from (4.9);  $V(k)$  is independent of  $m$ ;  $\lim_{m \rightarrow \infty} m = \infty$ . Therefore,

$$\lim_{m \rightarrow \infty} \Pi(m; k) = 1 - 0 - \infty = -\infty$$

■

Figure 4-6 depicts the profit as a function of the investment for different levels of initial knowledge ( $k$ ). For low initial values of  $k$ , the profit first goes down then up then back down. For medium values of  $k$ , it goes up then down. For high values of  $k$  it always goes down. These differing shapes are a consequence of the fact that it is very expensive to acquire additional information when  $k$  is high (see (4.9)). Since the value of knowing the customer perfectly is finite (recall (4.6)), for sufficiently high  $k$  the cost of acquiring additional far outweighs the benefits. The first order optimality conditions are

$$\frac{d\Pi(m_L; k)}{dm_L} = \frac{dV(L(m_L; k))}{dL} \frac{dL(m_L; k)}{dm_L} - 0 - 1 = 0$$

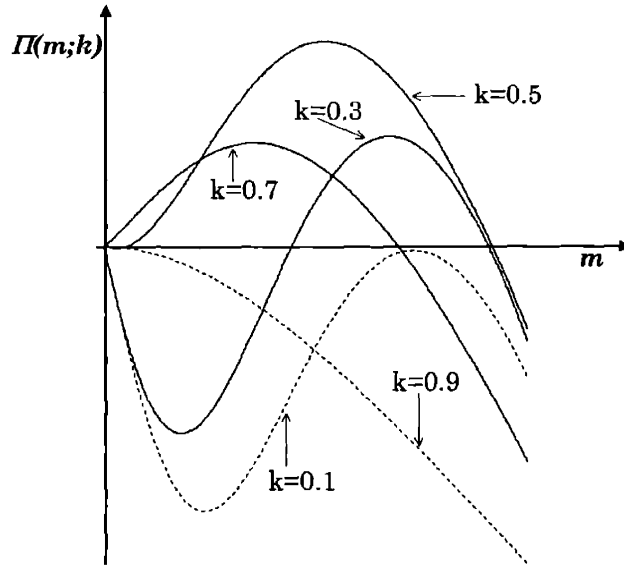


Figure 4-6: Profit from making an investment in learning of value  $m$  when the level of knowledge is  $k$

which can be simplified to

$$\frac{dV(L(m_L; k))}{dL} \frac{dL(m_L; k)}{dm_L} = 1.$$

The term  $\frac{dV(L(m_L; k))}{dL}$  is the slope of the value function, so it is strictly positive. The term  $\frac{dL(m_L; k)}{dm_L}$  is the slope of the learning function and it is also strictly positive. Since the learning function is concave it is possible that an extreme point of  $\Pi(m_L; k)$  is a local minimum. Therefore, it is important to look at the second order condition.

$$\frac{d^2 \Pi(m_L; k)}{dm_L^2} = \frac{d^2 V(L(m_L; k))}{dL^2} \frac{d(k, m_L)}{dm_L} + \frac{dV(L(m_L; k))}{dL} \frac{d^2 L(m_L; k)}{dm_L^2} < 0.$$

The only one of the four terms that can be negative is  $\frac{d^2 L(m_L; k)}{dm_L^2}$ . This gives us insights into the situations when the investment is not a local maximum. Of all the points that fulfill the first-order optimality conditions, the local maximum will be the one with the highest value of  $k$ . The characterization of the optimal policy follows from a series of results below.

**Lemma 21** For all  $k$  there is a level of investment  $m^H(k)$  above which it is unprofitable to invest.

**Proof.** The statement of the lemma can be restated as  $\forall k \exists m^H(k)$  s.t.  $\Pi(m^H; k) < 0 \forall m > m^H$ . This property follows immediately from  $\lim_{m \rightarrow \infty} (\Pi(m; k)) = -\infty$  ■

**Lemma 22** *If  $k_1 < k_2 < k_3$  then*

$$\Pi(L^{-1}(k_1, k_2), k_1) + \Pi(L^{-1}(k_2, k_3), k_2) = \Pi(L^{-1}(k_1, k_3), k_3)$$

**Proof.** LHS:  $\Pi(L^{-1}(k_1, k_2), k_1) + \Pi(L^{-1}(k_2, k_3), k_2) =$   
 $= [V(k_2) - V(k_1) - L^{-1}(k_1, k_2)] + [V(k_3) - V(k_2) - L^{-1}(k_2, k_3)]$   
 $= V(k_3) - V(k_1) - [L^{-1}(k_1, k_2) + L^{-1}(k_2, k_3)]$   
 By Lemma #,  $[L^{-1}(k_1, k_2) + L^{-1}(k_2, k_3)] = L^{-1}(k_1, k_3)$

Therefore the LHS can be simplified to

$$LHS = V(k_3) - V(k_1) - L^{-1}(k_1, k_3)$$

which is equal to the RHS by (4.14). ■

**Theorem 23** *If  $m^*(k_1) > 0$  and  $m^*(k_2) > 0$  are the optimal levels of investment given initial knowledge  $k_1$  and  $k_2$  respectively, then*

$$L(m^*(k_1); k_1) = L(m^*(k_2); k_2).$$

**Proof.** The theorem will be proved by demonstrating that assuming

$$L(m^*(k_1); k_1) \neq L(m^*(k_2); k_2)$$

leads to a contradiction in the optimality of  $m^*(k_1)$  and  $m^*(k_2)$ .

First, let  $L(m^*(k_1); k_1) = k_1^*$  and  $L(m^*(k_2); k_2) = k_2^*$ . Since  $L$  is a learning function it must satisfy (#), which taken with the statements  $m^*(k_1) > 0$  and  $m^*(k_2) > 0$  implies that  $k_1 < k_1^*$  and  $k_2 < k_2^*$ . Without loss of generality, assume that  $k_1 < k_2$ . If  $k_1^* \neq k_2^*$  then one of two statements is true: either (1)  $k_1^* < k_2^*$  or  $k_1^* > k_2^*$ . These two cases must be analyzed separately.

Case (1)  $k_1^* > k_2^*$

We have

From Lemma (21) we have

$$\Pi(L^{-1}(k_1, k_1^*), k_1) = \Pi(L^{-1}(k_1, k_2^*), k_1) + \Pi(L^{-1}(k_2^*, k_1^*), k_2^*)$$

If  $\Pi(L^{-1}(k_2^*, k_1^*), k_2^*) = 0$  then

$$\Pi(L^{-1}(k_1, k_1^*), k_1) = \Pi(L^{-1}(k_1, k_2^*), k_1),$$

which means that either  $L^{-1}(k_2^*, k_1^*) = 0$  (which would imply that  $k_1^* = k_2^*$ ) or that  $m^*(k_1) = L^{-1}(k_1, k_2^*)$ , contradicting the optimality of  $L^{-1}(k_1, k_1^*)$ .

If  $\Pi(L^{-1}(k_2^*, k_1^*), k_2^*) > 0$  then

$$\Pi(L^{-1}(k_2, k_1^*), k_2) = \Pi(L^{-1}(k_2, k_2^*), k_2) + \Pi(L^{-1}(k_2^*, k_1^*), k_2^*),$$



implies that

$$\Pi(L^{-1}(k_2, k_1^*), k_2) > \Pi(L^{-1}(k_2, k_2^*), k_2),$$

contradicting the optimality of  $k_2^*$ .

If  $\Pi(L^{-1}(k_2^*, k_1^*), k_2^*) < 0$  then

$$\Pi(L^{-1}(k_1, k_2^*), k_1) > \Pi(L^{-1}(k_1, k_1^*), k_1)$$

contradicting the optimality of  $k_1^*$ .

Case (2)  $k_1^* < k_2^*$

Applying Lemma 22 we obtain

$$\Pi(L^{-1}(k_2, k_2^*), k_2) = \Pi(L^{-1}(k_2, k_1^*), k_2) + \Pi(L^{-1}(k_1^*, k_2^*), k_1^*)$$

If  $\Pi(L^{-1}(k_1^*, k_2^*), k_1^*) = 0$  then

$$\Pi(L^{-1}(k_2, k_2^*), k_2) = \Pi(L^{-1}(k_2, k_1^*), k_2),$$

which means that either  $L^{-1}(k_1^*, k_2^*) = 0$  (which would imply that  $k_1^* = k_2^*$ ) or that  $m^*(k_2) = L^{-1}(k_2, k_1^*)$ , contradicting the optimality of  $L^{-1}(k_2, k_2^*)$ .

If  $\Pi(L^{-1}(k_1^*, k_2^*), k_1^*) > 0$  then

$$\Pi(L^{-1}(k_1, k_2^*), k_1) = \Pi(L^{-1}(k_1, k_1^*), k_1) + \Pi(L^{-1}(k_1^*, k_2^*), k_1^*),$$

implies that

$$\Pi(L^{-1}(k_1, k_2^*), k_1) > \Pi(L^{-1}(k_1, k_1^*), k_1)$$

contradicting the optimality of  $k_1^*$ .

If  $\Pi(L^{-1}(k_1^*, k_2^*), k_1^*) < 0$  then

$$\Pi(L^{-1}(k_2, k_2^*), k_2) < \Pi(L^{-1}(k_2, k_1^*), k_2)$$

which cannot be true because it contradicts the optimality of  $k_2^*$ . ■

Theorem 23 can be interpreted that the optimal information acquisition policy is to invest until a certain level of information is attained. Corollary 24 relates this level of information to the maximum level of information at which it is profitable to invest at all, from Lemma 21.

**Corollary 24** *If  $m^*(k) > 0$  then  $L(m^*(k); k) = k^H$*

**Proof.** From Theorem 23 we have  $L(m^*(k^H - \varepsilon); k^H - \varepsilon) = L(m^*(k); k)$ . Taking the limit

$$\lim_{\varepsilon \rightarrow 0} [L(m^*(k^H - \varepsilon); k^H - \varepsilon)] = k^H$$

yields the desired result. ■

The last few results have analyzed the investment policy when the initial knowledge is high. Now we turn to the situation when the initial level of knowledge is low.

**Lemma 25** *If  $\Pi(L^{-1}(0, k^H), 0) < 0$  then there exists a level of knowledge  $k^L$  such that it is never profitable to invest in learning if  $k \leq k^L$ .*

**Proof.** By the definition of the profit function,  $\Pi(L^{-1}(0, k^H), 0) < 0$  is equivalent to

$$V(k^H) - V(0) - L^{-1}(0, k^H) < 0.$$

Consider the profit function when the initial knowledge is a small  $\varepsilon > 0$ :

$$\Pi(L^{-1}(\varepsilon, k^H), \varepsilon) = V(k^H) - V(\varepsilon) - L^{-1}(\varepsilon, k^H)$$

Since

$$\Pi(L^{-1}(0, \varepsilon), 0) = V(\varepsilon) - V(0) - L^{-1}(0, \varepsilon)$$

we can use Lemma 22 to rewrite this profit function as

$$\begin{aligned} \Pi(L^{-1}(\varepsilon, k^H), \varepsilon) &= \Pi(L^{-1}(0, k^H), 0) - \Pi(L^{-1}(0, \varepsilon), 0) \\ &= [V(k^H) - V(\varepsilon)] - [L^{-1}(0, k^H) - L^{-1}(0, \varepsilon)] \end{aligned}$$

For sufficiently small  $\varepsilon$ ,  $[V(k^H) - V(\varepsilon)] \approx [V(k^H) - V(0)]$  and  $L^{-1}(0, \varepsilon) \approx 0$  which implies that

$$\Pi(L^{-1}(\varepsilon, k^H), \varepsilon) < 0.$$

Letting  $k^L$  be the highest  $\varepsilon$  that fulfills the condition imposed by the equation above yields the result of the Lemma. ■

When the conditions of Lemma 25 are met, it is not optimal to invest in information acquisition. This is likely to happen when learning is expensive (the method of learning is inefficient) or when the slope of the value function is low when knowledge is near 0. Issues of sensitivity analysis with respect to the problems parameters will be dealt with in §4.5.3.

The optimal level of investment as function of the initial knowledge is depicted in Figure 4-7. The most salient feature of this curve is that it is discontinuous at  $k^L$ . This is because the conditions of Lemma 25 are satisfied in this case, and the optimal level of investment is zero.

The profit from investment when investment is optimal is depicted in Figure 4-8a. The percentage increase in the value of the customer ( $\frac{\Pi(m^*(k), k)}{V(k)}$ ) as a function of the initial knowledge is depicted in Figure 4-8b. Finally, the return on investment ( $\frac{\Pi(m^*(k), k)}{m^*(k)}$ ) is depicted in Figure 4-8c.

**Constrained Resources** Suppose that the firm cannot invest more than  $\hat{m}$  on learning.  $k_{\hat{m}}^L$ .

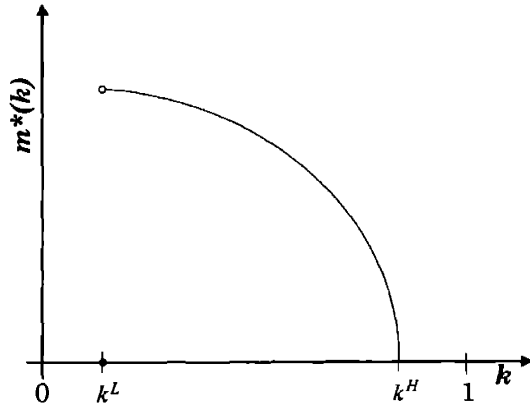


Figure 4-7: Optimal level of investment as a function of knowledge

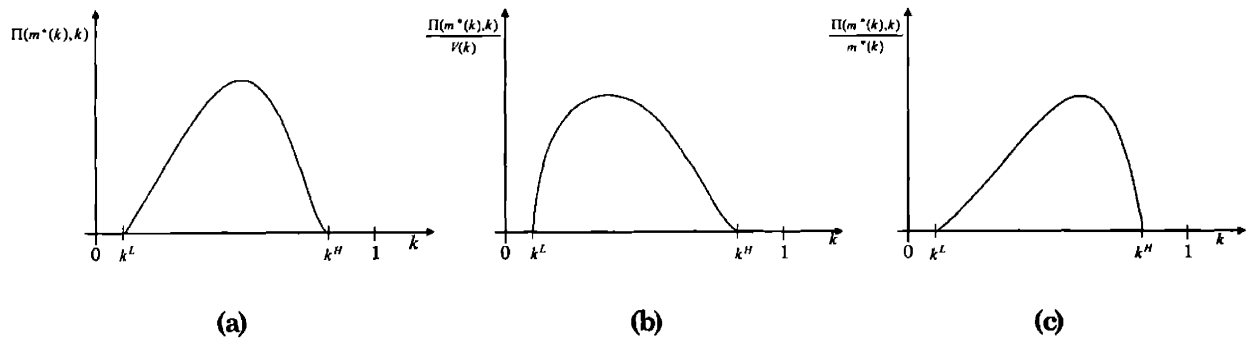


Figure 4-8: Profit function, relative increase in customer value, and return on investment when investment is made optimally

The optimization problem becomes

$$\Pi(m_L; k) = V(L(m_L; k)) - V(k) - m_L$$

s.t.

$$m_L \leq \hat{m}.$$

The optimality conditions are

$$\frac{d\Pi(m_L; k)}{dm_L} = \mu$$

and

$$\mu(m - \hat{m}) = 0$$

where  $\mu$  is a Lagrange multiplier. The case when the constraint is not binding is equivalent to the unconstrained optimization case, therefore the analysis in this section will focus on the case when  $\mu > 0$ .

**Theorem 26** *There is a level of knowledge  $k_{\min}$  such that, for all  $k$ , if the firm is unable to reach that level of knowledge then it is better not to invest at all.*

**Proof.** Let  $k_{\min} = L(m^*(k_m^L); k_m^L)$ , where  $k_m^L$  is the lowest level of knowledge at which it is profitable to invest. If  $k_{\min} = k^H$ , then the constraint is never binding and the solution will be the same as in the unconstrained problem. If the constraint is binding, i.e.,  $\mu(k_m^L) > 0$ , then  $m^*(k_m^L) = \hat{m}$ , and for small  $\varepsilon$ ,  $m^*(k_m^L + \varepsilon) = \hat{m}$ . Since  $(k_m^L + \varepsilon) > k_m^L$  it must be the case that  $L(\hat{m}; k_m^L + \varepsilon) > L(\hat{m}; k_m^L)$ . ■

The theorem above implies that when there is a maximum limit to the investment that can be made, firms will invest less often than they would in the unconstrained problem. This result is formalized in the Corollary below.

**Corollary 27** *There are situations where it is optimal to invest in the constrained problem not in the unconstrained problem even when it would be feasible to make the investment..*

**Proof.** Follows immediately from  $k^L < k_m^L$  and Theorem 26. ■

Figure depicts the optimal investment levels as a function of knowledge. In  $[0, k^L]$  and in  $[k^H, 1]$  the constraint is not binding. In region  $[k^L, k_m^L]$  the constraint is binding but no investment is made. In this region, if the firm could have invested any amount it would have invested  $m^* > \hat{m}$ , but if it is constrained to invest at most  $\hat{m}$  then it is better not to invest at all! Figure 4-10 shows  $\Pi(m; k)$  for  $k = k_m^L$  and  $k = k^L$  to help explain this phenomenon.

Figure 4-11 depicts the profit assuming investment was made optimally as a function of knowledge.

**Theorem 28** *Consider a firm that faces the problem of investing up to  $\hat{m}$  in order to learn about a customer currently at knowledge level  $k$ . The optimal investment policy can be summarized by two simple rules: (1) if  $k < k_m^L$  or  $k > k^H$ , do not invest; (2) otherwise, invest  $\min(\hat{m}, L^{-1}(k, k^H))$*

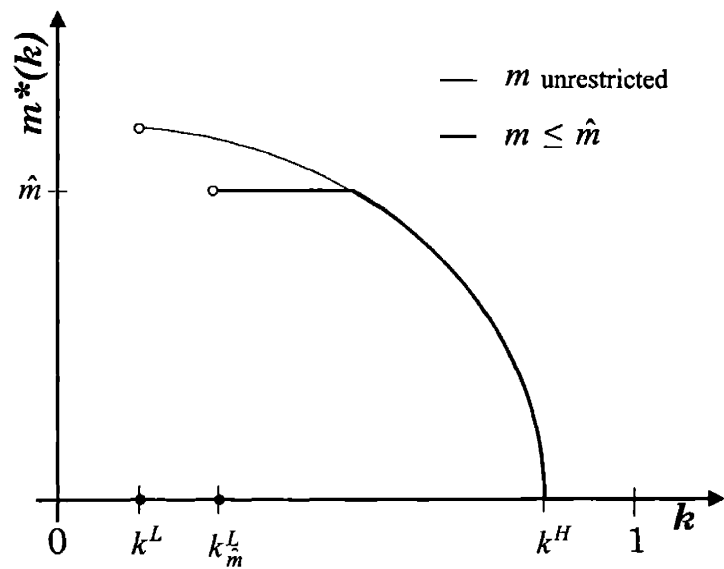


Figure 4-9: Optimal level of investment under constraints

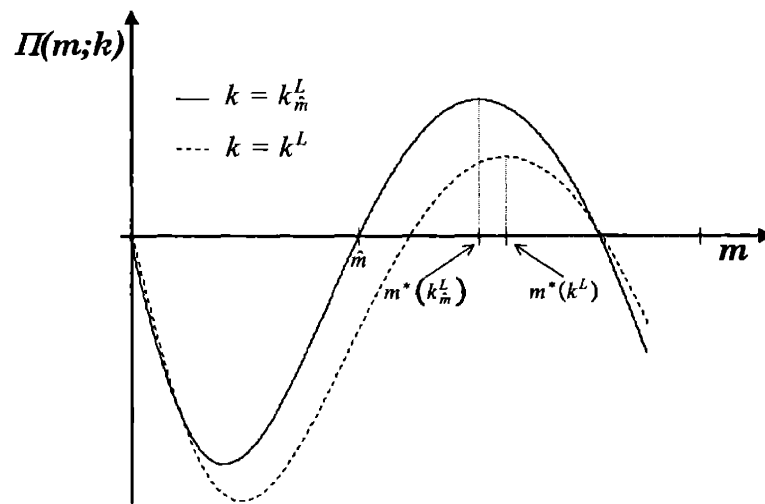


Figure 4-10: Understanding the discontinuity in  $m^*(k)$

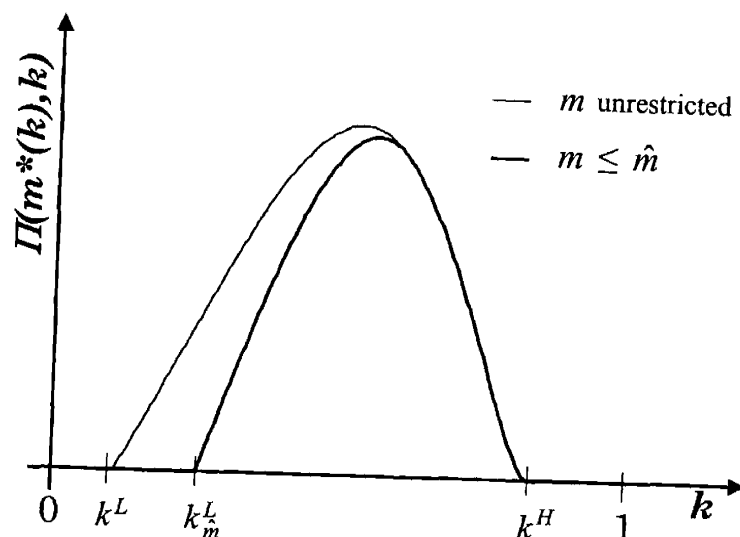


Figure 4-11: Profit as a function of investment under constrained resources

**Proof.** Not investing when if  $k < k_{\hat{m}}^L$  follows from Lemma 25. Not investing when if  $k > k^H$  follows from Lemma 21. Investing  $\min(\hat{m}, L^{-1}(k, k^H))$  follows from Corollary 24. ■

### Two customers, constant acquisition rate

This section analyzes the situation when a company has a fixed amount of resources,  $\hat{m}_L$ , that must be allocated to learning about two different customers. The profit function is

$$\Pi(m_{L_1}, m_{L_2}; k_1, k_2) = V(L(k_1, m_{L_1})) - V(L(k_1)) + V(L(k_2, m_{L_2})) - V(L(k_2)) - m_L$$

where the total resources allocated to learning are given by

$$m_{L_1} + m_{L_2} = m_L \leq \hat{m}_L.$$

Letting

$$\tilde{m}_L = \max(\hat{m}_L, [m^*(k_1) + m^*(k_1)])$$

and substituting

$$m_{L_2} = \tilde{m}_L - m_{L_1}$$

into the profit function yields

$$\Pi(m_{L_1}; k_1, k_2) = V(L(k_1, m_{L_1})) - V(k_1) + V(L(k_2, \tilde{m}_L - m_{L_1})) - V(k_2) - \hat{m}_L,$$

which is a function of only one unknown variable and can be solved using the same techniques

as in the previous section.

The KKT optimality conditions are

$$\frac{d\Pi(m_L; k_1, k_2)}{dm_L} = \frac{dV(L(k_1, m_L))}{dL} \frac{dL(k_2, m_L)}{dm_L} + \frac{dV(L(k_2, m_L))}{dL} \frac{dL(k_2, m_L)}{dm_L} - 1 = \mu$$

and

$$\mu(m_L - \hat{m}_L) = 0$$

The theorem below describes the optimal control policy, which follows immediately from the insights obtained in §4.5.2.

**Theorem 29** *Let  $k^1$  and  $k^2$  denote the level of knowledge about the first and the second customer respectively. The optimal policy to invest  $\hat{m}_L$  to learn about two customers is to follow four steps: (1) Do not invest in a customer with  $k < k_{\hat{m}_L}^L$  or  $k > k^H$ . (2) Invest on the customer with the highest  $\frac{\partial \Pi}{\partial m}$  until  $\frac{\partial \Pi}{\partial m} \Big|_{k=k^1} = \frac{\partial \Pi}{\partial m} \Big|_{k=k^2}$ . (3) If  $k^1 \neq k^2$  then invest in the customer with the lower  $k$  until  $k^1 = k^2$ . (4) If  $k^1 = k^2$  invest equally in both customers as long as  $\frac{\partial \Pi}{\partial m} > 0$ .*

**Proof.** Step (1) is an immediate consequence of Theorem 28. Step (2) follows from the fact that the profit from investing  $\Delta m$  increases with  $\frac{\partial \Pi}{\partial m}$ , i.e.,

$$\Pi(\Delta m; k^i) \approx \Delta m \left. \frac{\partial \Pi}{\partial m} \right|_{k=k^i}$$

and therefore

$$\Pi(\Delta m; k^i) > \Pi(\Delta m; k^j)$$

if and only if

$$\left. \frac{\partial \Pi}{\partial m} \right|_{k=k^i} > \left. \frac{\partial \Pi}{\partial m} \right|_{k=k^j}$$

Step (3) follows from the fact that if  $\left. \frac{\partial \Pi}{\partial m} \right|_{k=k^1} = \left. \frac{\partial \Pi}{\partial m} \right|_{k=k^2}$  and  $k^1 \neq k^2$  then  $\left. \frac{\partial^2 \Pi}{\partial m^2} \right|_{k=k^1} > 0$  and  $\left. \frac{\partial^2 \Pi}{\partial m^2} \right|_{k=k^2} < 0$ . Step (4) follows from the fact that investing money when  $\frac{\partial \Pi}{\partial m}$  always generates negative profits and is therefore always undersirable. ■

## **$n$ Customers, constant acquisition rate**

This section generalizes the problem of §4.5.2 to the case when a budget of size  $\hat{m}_L$  must be allocated to learning about  $n$  different customers. The profit function becomes

$$\Pi(m_{L_1}, m_{L_2}, \dots, m_{L_n}; k_1, k_2, \dots, k_n) = \sum_{i=1}^n [V(L(k_n, m_{L_n})) - V(k_n)] - m_L$$

where

$$m_L = \sum_{i=1}^n [m_{L_n}]$$

subject to the constraint

$$\sum_{i=1}^n [m_{L_n}] \leq \hat{m}_L.$$

The KKT optimality conditions are

$$\frac{\partial \Pi}{\partial m_{L_i}} = \mu \text{ for } i = 1, \dots, n$$

and

$$\mu \left( \sum_{i=1}^n [m_{L_n}] - \hat{m}_L \right) = 0.$$

The Lagrange multiplier  $\mu$  will be zero if the constraint is not binding. Since this problem is a direct generalization of §4.5.2, the optimal control policy will be a generalization of the policy described in Theorem 29.

**Theorem 30** *Let  $k^i$  denote the level of knowledge about the  $i$ 'th customer. The optimal policy to invest  $\hat{m}_L$  to learn about  $n$  customers is to follow five steps: (1) Do not invest in a customer with  $k < k_{\hat{m}_L}^L$  or  $k > k^H$ . (2) Rank all customers in descending order of  $\frac{\partial \Pi}{\partial m}$ , so that  $\frac{\partial \Pi}{\partial m} \Big|_{k=k^i} > \frac{\partial \Pi}{\partial m} \Big|_{k=k^j}$  as long as  $i < j$ . (3) Invest in customer 1 until  $\frac{\partial \Pi}{\partial m} \Big|_{k=k^1} = \frac{\partial \Pi}{\partial m} \Big|_{k=k^2}$  (4) Invest in customers 1 and 2 until  $\frac{\partial \Pi}{\partial m} \Big|_{k=k^1=k^2} = \frac{\partial \Pi}{\partial m} \Big|_{k=k^3}$  (5) Repeat step 4  $\frac{\partial \Pi}{\partial m} \Big|_{k=k^i} = \frac{\partial \Pi}{\partial m} \Big|_{k=k^j}$  for all  $i, j$ . (6) Invest equally in all customers as long as  $\frac{\partial \Pi}{\partial m} > 0$ .*

**Proof.** This theorem is a direct generalization of the policy in Theorem 29, and it is optimal for the same reasons described in the proof of that theorem. ■

### Zero customers, variable acquisition rate

The customer acquisition rate depends on the intensity of advertising, as described in §4.5.1. We make the assumption that firms do not know anything about the preferences of new customers, so the value of new customers is  $V(0)$ .

Suppose that the firm makes an investment of  $m_A$  in advertising. The expected profit from this investment is given by

$$E_n \Pi(m_A) = nV(0) - m_A$$

We know from (4.13) that the customer arrival process is Poisson with rate  $\lambda(m_A)$ , so we can write

$$E_n (\Pi(m_A)) = \lambda(m_A) V(0) - m_A.$$



The function  $\lambda(m_A)$  will be defined as

$$\lambda(m_A) = c_1 + c_2 \ln(m_A).$$

This function is depicted in Figure 4-12. According to Rao (REF), this function agrees with empirical results in the usual range of advertising expenditures.

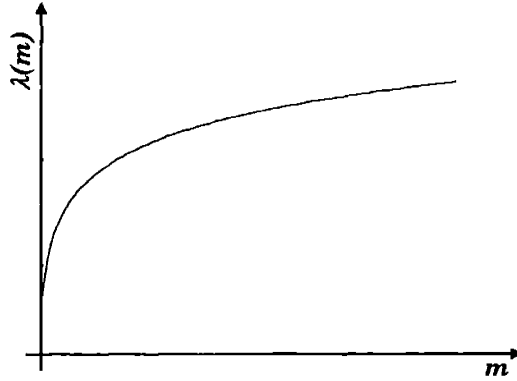


Figure 4-12: Customer acquisition rate as a function of investment

Furthermore, it satisfies the properties described in 4.5.1, as proved below.

**Claim 31** *The function*

$$\lambda(m_A) = c_1 + c_2 \ln(m_A)$$

*where  $c_1$  and  $c_2$  are positive constants, satisfies the properties described in 4.5.1.*

**Proof.**

$$\frac{\partial \lambda(m_A)}{\partial m_A} = \frac{\partial}{\partial m_A} (c_1 + c_2 \ln(m_A)) > 0$$

$$\lim_{m_A \rightarrow \infty} \frac{\partial^2 \lambda(m_A)}{\partial m_A^2} < 0.$$

■

The optimization problem can now be formulated as

$$\max_{m_A} E(\Pi(m_A)) = [c_1 + c_2 \ln(m_A)] V(0) - m_A \quad (4.15)$$

subject to

$$m_A \leq \hat{m}_A.$$

Optimality conditions are

$$\frac{d\Pi(m_A)}{dm_A} = \mu,$$

and

$$\mu(m - \hat{m}) = 0.$$

The profit function in (4.15) is depicted in Figure 4-11. Since the profit function is concave, it is unnecessary to check second order conditions because we can be assured that the second derivative will be negative any extreme points are certain to be maxima.

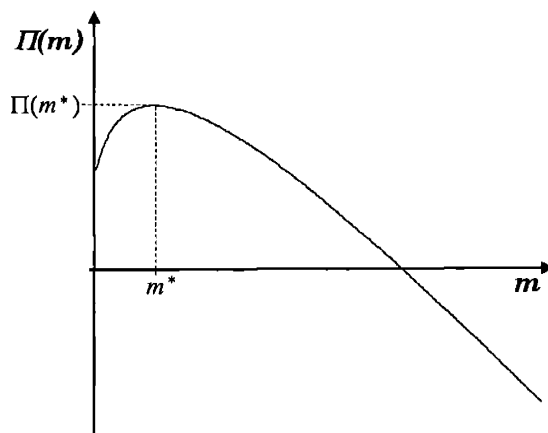


Figure 4-13: Profitability of different levels of investment in customer acquisition

### ***n* Customers, variable acquisition rate**

This section combines the results of §4.5.2 and §4.5.2 in order to analyze the general problem faced by a company with a fixed budget that can be used to invest in learning or customer acquisition. The profit function is given by

$$\Pi(m_L, m_A; k) = \sum_{i=1}^n [V(L(k_n, m_{L_n})) - V(k_n)] + cV(0) - (m_A + m_L)$$

subject to

$$m_A + m_L \leq \hat{m}$$

and

$$m_L = \sum_{i=1}^n [m_{L_n}].$$

The KKT optimality conditions are

$$\frac{\partial \Pi}{\partial m_{L_i}} = \mu, \text{ for } i = 1, \dots, n$$

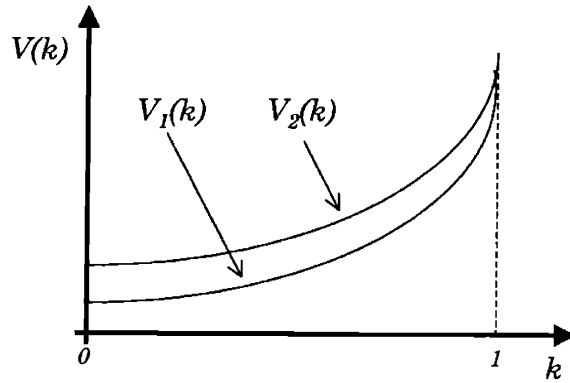


Figure 4-14: Value functions corresponding to different interaction strategies

and

$$\mu \left( m_A + \sum_{i=1}^n [m_{L_n}] - \hat{r}_n \right) = 0$$

The optimal investment policy is given in the theorem below. It extends the result of Theorem by including an additional  $\frac{\partial \Pi}{\partial m}$  index for investment in advertising.

**Theorem 32** Let  $k^i$  denote the level of knowledge about the  $i$ 'th customer. The optimal policy to invest  $\hat{r}_L$  to learn about  $n$  customers is to follow five steps: (1) Do not invest in a customer with  $k < k_{\hat{r}_L}^L$  or  $k > k^H$ . (2) Rank all customers in descending order of  $\frac{\partial \Pi}{\partial m}$ , so that  $\frac{\partial \Pi}{\partial m} \Big|_{k=k^i} > \frac{\partial \Pi}{\partial m} \Big|_{k=k^j}$  as long as  $i < j$ . (3) Calculate  $\frac{\partial \Pi}{\partial m}$  for investment in advertising and include this index in the ranking in (2) as if it was the  $(n+1)$ st customer. (4) Invest in customer 1 until  $\frac{\partial \Pi}{\partial m} \Big|_{k=k^1} = \frac{\partial \Pi}{\partial m} \Big|_{k=k^2}$  (5) Invest in customers 1 and 2 until  $\frac{\partial \Pi}{\partial m} \Big|_{k=k^1=k^2} = \frac{\partial \Pi}{\partial m} \Big|_{k=k^3}$  (6) Repeat step 5  $\frac{\partial \Pi}{\partial m} \Big|_{k=k^i} = \frac{\partial \Pi}{\partial m} \Big|_{k=k^j}$  for all  $i, j$ . (7) Invest equally in all customers as long as  $\frac{\partial \Pi}{\partial m} > 0$ .

**Proof.** This theorem is a direct generalization of the policy in Theorem 30, and it is optimal for the same reasons described in the proof of that theorem. ■

### 4.5.3 Sensitivity Analysis

#### Quality of recommendation policy

In this section we demonstrate how different interaction policies can affect investment strategies. Most companies that customize their services online use myopic recommendation policies. These companies often make significant investments in learning about their customers. If they improve their customization strategy by taking learning into account, how should their investment strategy change? Figure 4-14 shows the value functions corresponding to a good recommendation strategy ( $V_2$ ) and a bad recommendation strategy ( $V_1$ ).

The parametrized profit function is

$$\Pi(m_L, m_A; k, a) = \sum_{i=1}^n [V_r(L(k_n, m)) - V(k_n)] + wV(0) - (m_A + m_L), \quad r = 1, 2$$

subject to

$$m_A + m_L \leq \hat{m}$$

where

$$m_L = \sum_{i=1}^n [m_{L_n}].$$

Figure 4-15 shows the optimal investment levels for the two different value functions. Figure 4-16 shows the profit corresponding to these investments.

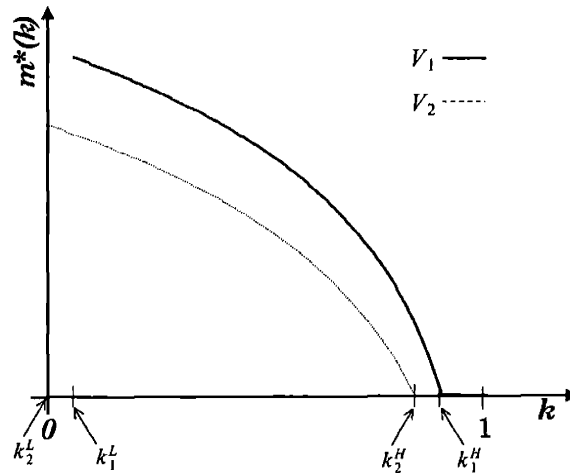


Figure 4-15: Optimal investment levels for two different value functions

### Effectiveness of Learning Method

Consider the situation where the company has the opportunity to use a questioning method that learns faster but is costly to implement. An example might be calling the customer over the telephone in order to learn about them or use a questionnaire sent over mail (the telephone call costs more but yields more information). How will the new learning system affect the profit and the investment policy? Which system should be used?

The first step in performing this sensitivity analysis is to define a suitable learning function that allows us to parametrize the effectiveness of the learning method.

**Claim 33** *The function*

$$L(m_L; k) = 1 + (k - 1) a^{m_L} \quad (4.16)$$

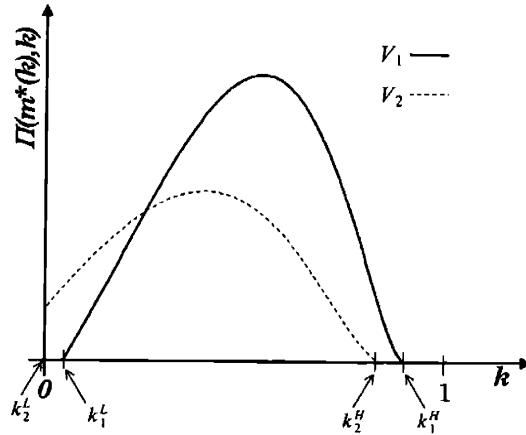


Figure 4-16: Profit given optimal investment for two different value functions

where  $a \in (0, 1)$  satisfies assumptions (4.7), (4.8), (4.9), and (4.10), making it a suitable Learning function.

**Proof.** (A1):  $L(0; k) = k$  for all  $k$

$$L(0; k) = 1 + (k - 1)a^0 = 1 + (k - 1)1 = k$$

(A2)  $\frac{\partial L(m; k)}{\partial m} > 0$

$$\frac{\partial L(m; k)}{\partial m} = (k - 1)a^m \ln a$$

$$a \in (0, 1) \rightarrow \ln a < 0$$

$$k \in [0, 1) \rightarrow (k - 1) < 0$$

$$a^m > 0$$

Therefore,  $\frac{\partial L(m; k)}{\partial m} > 0$

(A3)  $\lim_{m \rightarrow \infty} L(m; k) = 1$

$$\lim_{m \rightarrow \infty} (1 + (k - 1)a^m) = 1 + (k - 1) \lim_{m \rightarrow \infty} a^m = 1 + (k - 1)0 = 1$$

(A4)  $L(m_2; L(m_1; k)) = L(m_1 + m_2; k)$

$$L(m_2; L(m_1; k)) = 1 + ((1 + (k - 1)a^{m_1}) - 1)a^{m_2}$$

$$= 1 + ((k - 1)a^{m_1})a^{m_2}$$

$$= 1 + (k - 1)a^{m_1 + m_2} = L(m_1 + m_2; k) \quad \blacksquare$$

The function (4.16) is particularly appropriate to describe the learning process because in addition to possessing the necessary features the parameter  $a$  has an interesting managerial interpretation. It is the effectiveness of the learning method. The smaller the value of  $a$ , the more effective the method of learning. Two learning functions are depicted in Figure 4-17. In this case,  $L_2$  is more efficient than  $L_1$ .

The parametrized profit function is

$$\Pi(m_L, m_A; k, a) = \sum_{i=1}^n [V(1 + (k_n - 1)a_1^{m_L}) - V(k_n)] + wV(0) - (m_A + m_L), \quad r = 1, 2$$

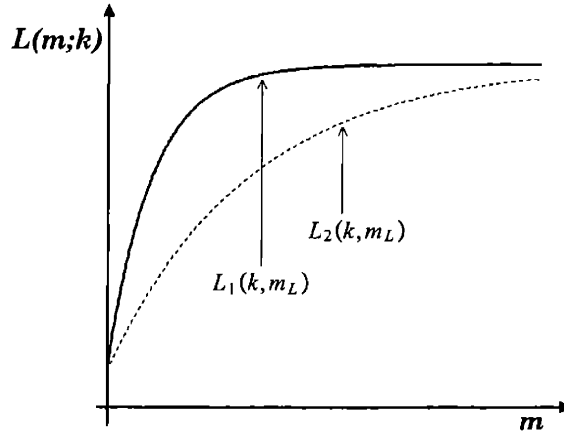


Figure 4-17: Learning policies of different effectiveness

subject to

$$m_A + m_L \leq \hat{m}$$

where

$$m_L = \sum_{i=1}^n [m_{L_n}]$$

The optimal profit function can be written as a function of the parameter  $a$ :

$$\Pi^*(a) = \Pi(m_L(a), m_A(a); k, a).$$

Differentiating both side with respect to  $a$  and simplifying the result yields

$$\frac{d\Pi^*(a)}{da} = \frac{\partial \Pi(m_L(a), m_A(a); k, a)}{\partial a} \Big|_{m_L=m_L(a), m_A=m_A(a)}$$

Figure 4-18a shows the optimal investment levels for learning functions  $L_1$  and  $L_2$ . In this figure,  $L_1$  (represented by the solid line) is the better learning function. Figure 4-18b shows the profit when investment is made optimally for learning functions  $L_1$  and  $L_2$ . There are two important results from this analysis. First, if  $a_1 < a_2$  then  $k_1^L < k_2^L$  and  $k_1^H > k_2^H$ . Second, if  $a_1 < a_2$  then  $\Pi^*(m_L; k, a_1) > \Pi^*(m_L; k, a_1)$  for all  $k$ .

## 4.6 Discussion

The analysis in §4.4 provides the theoretical framework for a decision-support support tool that can be used to determine whether firms should provide service or ask questions. Having the flexibility to ask questions at any point in time is better than the usual practice of asking new customers to answer questionnaires and always provide service after that for two main

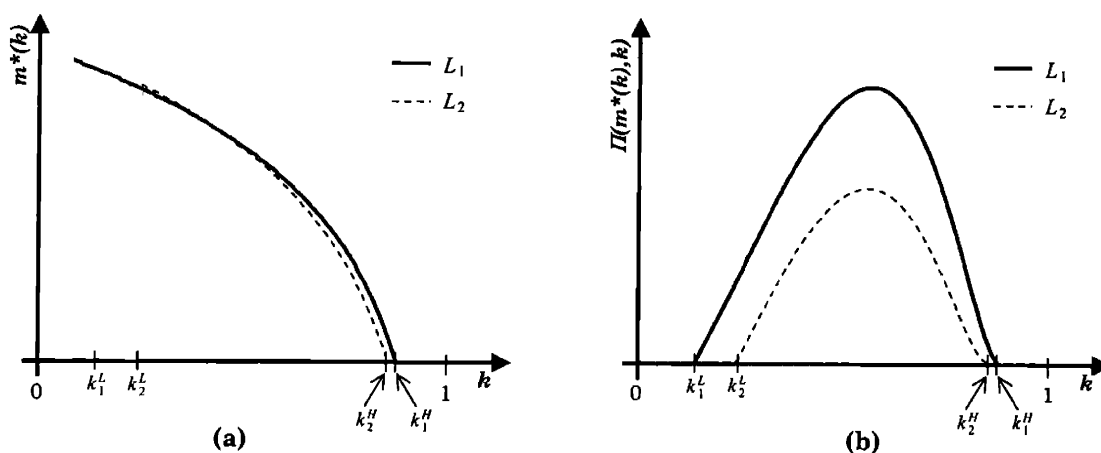


Figure 4-18: Optimal investments and their corresponding profit for different learning functions

reasons. First, because firms can ask fewer questions at the beginning of the relationship and start providing service sooner. Second, because if a customer behaves in an unexpected manner the firm is often better off asking a question rather than continuing trying to learn through observations.

The learning function defined in § 4.5, based on four simple and non-controversial assumptions leads to a number of insights into the value of information about customers' preferences. First, that information is not most valuable when companies do not know anything about the customer and neither is it most valuable when firms know the customer well. It is most valuable somewhere in the middle, as shown in Figure 4-8. This result was found by understanding the interaction between a concave learning function and a convex value function. When firms don't know anything about their customers, information is cheap but it does not lead to significant changes in the value of the customer. When the firm already knows the customer well and additional piece of information is very valuable, but it is also very expensive to attain. The second important insight is that if the firm is unable to make a significant investment in information acquisition then it is better not to invest at all. This result, depicted in Figure 4-9, is a consequence of the fact that a minimum level of knowledge must be achieved in order for the transaction to be profitable. Finally, as proved in Theorem 28, we note that the optimal information acquisition policy can be expressed in very simple terms: invest only when the initial level of knowledge falls between two given thresholds, and do so until the level of knowledge reaches the higher of the two thresholds..

The policy derived in § 4.5.2 captures the tradeoff between customer acquisition and customer retention. This is an important building block in trying to obtain a fuller picture of the financial aspects of providing services through automated interfaces. Information is always desirable if it is free, but in the real world firms must weight the cost of acquiring

information with other budgetary possibilities. The alternative investment considered here is increasing advertising expenditure. The optimal level of advertising has been extensively studied in the literature, and is therefore suitable as a point of comparison for investing in information acquisition.

The sensitivity analysis in § 4.5.3 shows that firms that adopt new technologies should change their information acquisition policies accordingly. Two changes must be made if firms improve their interaction policies (i.e., obtain a better value function). First, the threshold at which it is profitable to acquire information will be lower. This is because a better interaction policy means that the firm has a greater ability to learn through interactions, and therefore a small amount of information is sufficient to cause a significant improvement in the value function. By contrast, if a firm uses a myopic interaction policy then a small improvement in knowledge might not lead to any changes at all in the service to be recommended. Second, the threshold at which firms should stop purchasing information is also lower. This change is a direct consequence of the fact that value functions corresponding to bad recommendation policies have very steep slopes as firms approach perfect information. Firms that improve the learning acquisition policy must also make changes in their information acquisition policies.

The results obtained in this chapter raise a number of further research questions. First, how do information acquisition policies change if the firm has the option of purchasing information at any point in the relationship? This is essentially combining the models of §4.4 and §4.5. Second, how can the methodology of §4.5 be modified to account for non-symmetric value functions? The analysis conducted here assumes that all customers have the same value if the firm knows them perfectly, which is not the case in many real-world applications. Finally, there are a number of other applications in services operations management (e.g., pricing) where the learning function defined § 4.5.1 in may be useful.



# Bibliography

- [1] P.S. Adler and K.B. Clark, *Behind the learning curve: A sketch of the learning process*, Management Science **37** (1991), no. 3, 267–281.
- [2] A.A. Alchian, *Costs and outputs*, Stanford University Press, 1959.
- [3] L. Argote and D Epple, *Learning curves in manufacturing*, Science **247** (1990), no. 23, 920–924.
- [4] D. Ariely, *The personal aspects of electronic agents*, Working paper, MIT Sloan School of Management, Cambridge, MA, USA, 2001.
- [5] H. Asher, *Cost quantity relationships in airframe production*, Technical report r-291, RAND Corp., Santa Monica, CA, USA, 1956.
- [6] K.J. Astrom, *Optimal control of markov decision processes with incomplete state estimation*, Journal of Mathematical Analysis and Applications **10** (1965), 174–205.
- [7] A.B. Berghell, *Production engineering in the aircraft industry*, McGraw-Hill, 1944.
- [8] D. Bertsekas, *Dynamic programming and optimal control, volume 1*, Athena Scientific, 1995.
- [9] ———, *Dynamic programming and optimal control, volume 2*, Athena Scientific, 1995.
- [10] D. Bertsekas and J. Tsitsiklis, *An analysis of stochastic shortest path problems*, Mathematics of Operations Research **16** (1991), no. 3, 580–595.
- [11] D. Bertsekas and J. Tsitsiklis, *Neuro-dynamic programming*, Athena Scientific, Belmont, MA, 1996.
- [12] G.R. Bitran and M. Lojo, *A framework for analyzing service operations*, European Management Journal **11** (1993), no. 3, 271–282.
- [13] G.R. Bitran and M. Lojo, *A framework for analyzing the quality of the customer interface*, European Management Journal **11** (1993), no. 4, 385–396.
- [14] G.R. Bitran and S. Mondschein, *Managing the tug-of-war between supply and demand in the service industries*, European Management Journal **15** (1997), no. 5, 523–536.

- [15] D. Blackwell, *Discounted dynamic programming*, Annals of Mathematical Statistics **36** (1965), no. 1, 226–235.
- [16] C. Boutilier and D. Poole, *Computing optimal policies for partially observable decision processes using compact representations*, Proceedings of the Thirteenth National Conference on Artificial Intelligence (1996), 1168–1175.
- [17] S. Brumelle and K. Sawaki, *Generalized policy improvement for simple dynamic programs with an application to partially observable markov decision processes*, Working paper 546, Faculty of Commerce, University of British Columbia, British Columbia, Canada, 1978.
- [18] J.E. Carrillo and C. Gaimon, *Improving manufacturing performance through process change and knowledge creation*, Management Science **46** (2000), no. 2, 265–288.
- [19] A.R. Cassandra, *Optimal policies for partially observable markov decision processes*, Technical report, Brown University Department of Computer Science, Providence, R.I., 1995.
- [20] Kaelbling L.P. Cassandra, A.R. and M.L. Littman, *Acting optimally in partially observable stochastic domains*, Proceedings of the Twelfth National Conference on Artificial Intelligence (1994), 1023–1028.
- [21] Littman M.L. Cassandra, A.R. and N.L. Zhang, *Incremental pruning: A simple, fast, exact method for partially observable markov decision processes*, Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-97), Morgan Kaufmann Publishers, San Francisco, CA, 1997, pp. 54–61.
- [22] R. B. Chase and D.M. Stewart, *Make your service fail-safe*, Sloan Management Review **35** (1994).
- [23] H. Cheng, *Algorithms for partially observable markov decision processes*, Ph.d. thesis, University of British Columbia, British Columbia, Canada, 1988.
- [24] M.K. Clayton, *Covariate models for bernoulli bandits*, Sequential Analysis **8** (1989), 405–426.
- [25] M. Dada and R. Marcellus, *Process control with learning*, Operations Research **42** (1994), no. 2, 323–336.
- [26] E. Denardo, *Contraction mappings in the theory underlying dynamic programming*, SIAM Review **9** (1967), no. 2, 165–177.
- [27] P.M. Doney and J.P. Cannon, *An examination of the nature of trust in buyer-seller relationships*, Journal of Marketing **61** (1997), no. April, 35–51.

- [28] J.R. Dorroh, *Investment in knowledge: A generalization of learning by experience*, Management Science **40** (1994), no. 8, 947–958.
- [29] A.W. Drake, *Observation of a markov process through a noisy channel*, D.sc. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 1962.
- [30] J.N. Eagle, *The optimal search for a moving target when the search path is constrained*, Operations Research **32** (1984), no. 5, 1107–1115.
- [31] J.H. Eaton and L.A. Zadeh, *Optimal pursuit strategies in discrete state probabilistic systems*, Trans. ASME Ser. D.J. Basic Eng. **84** (1962), 23–29.
- [32] J.E. Eckles, *Optimum maintenance with incomplete information*, Operations Research **16** (1968), 1058–1067.
- [33] A. Federgruen and P. Zipkin, *An efficient algorithm for computing optimal  $(s,s)$  policies*, Operations Research **32** (1984), 1268–1285.
- [34] C. Fine, *Quality improvement and learning in productive systems*, Management Science **32** (1986), no. 10, 1301–1315.
- [35] ———, *A quality control model with learning effects*, Operations Research **36** (1988), no. 2, 437–444.
- [36] J.H. Gilmore and B.J. Pine II, *The four faces of mass customization*, Harvard Business Review **January-February** (1997), 91–101.
- [37] J.C. Gittins, *Bandit processes and dynamic allocation indices*, Journal of the Royal Statistical Society B **41** (1979), 148–164.
- [38] ———, *Multi-armed bandit allocation indices*, Wiley, Chichester, NY, 1989.
- [39] C.G. Gooley and J.M. Lattin, *Dynamic customization of marketing messages in interactive media*, Working paper, Stanford University Graduate School of Business, Stanford, CA, 1998.
- [40] William H. Greene, *Econometric analysis, 4th edition*, Prentice Hall, New Jersey, U.S.A., 2000.
- [41] T.R. Gullledge, M.M. Tarimcilar, and N.K. Womer, *Estimation problems in rate-augmented learning curves*, IEEE Transactions on Engineering Management **44** (1997), no. 1, 91–97.
- [42] E.A. Hansen and Z. Feng, *Dynamic programming for pomdps using a factored state representation*, Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems, Breckenridge, CO, 2000, pp. 130–139.

- [43] M. Hauskrecht, *Planning and control in stochastic domains with imperfect information*, Ph.d. dissertation, Massachusetts Institute of Technology, Cambridge, MA, US, 1997.
- [44] ———, *Value-function approximations for partially observable markov decision processes*, Journal of Artificial Intelligence Research **13** (2000), 33–94.
- [45] M. Hauskrecht, N. Meulau, L.P. Kaelbling, and C. Boutilier, *Hierarchical solution of markov decision processes using macro-actions*, Proceedings of the fourteenth conference on uncertainty in artificial intelligence, University of Wisconsin Business School, Madison, WI, USA, 1998.
- [46] L.P. Kaelbling, *Hierarchical learning in stochastic domains: Preliminary results*, Proceedings of the Tenth International Conference on Machine Learning, 1993, pp. 167–173.
- [47] ———, *Learning to achieve goals*, Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence Planning Systems, Morgan Kaufmann, Chambéry, France, 1993.
- [48] L.P. Kaelbling, M.L. Littman, and A.W. Moore, *Reinforcement learning: A survey*, Journal of Artificial Intelligence **4** (1996), 237–285.
- [49] J. Kakalik, *Optimum policies for partially observable markov systems*, Operations research center, technical report tr-18, Massachusetts Institute of Technology, Cambridge, MA, US, 1965.
- [50] Larson B. M. Katz, K. L. and R. C. Larson, *Prescription for the waiting-in-line blues: Entertain, enlighten, and engage*, Sloan Management Review **32** (1991), no. 2, 44–53.
- [51] M.R. Killingsworth, *Learning by doing and investment in training: A synthesis of two rival models of the life cycle*, Review of Economic Studies **49** (1982), 263–271.
- [52] P. Kollock, *The production of trust in online markets*, Advances in Group Processes **16** (1999), 99–123.
- [53] D.E. Lane, *A partially observable model of decision making by fishermen*, Operations Research **37** (1989), 240–254.
- [54] T. Levitt, *Production-line approach to service*, Harvard Business Review **50** (1972), no. September-October.
- [55] R.J. Lewicki, D.J. McAllister, and R.J. Bies, *Trust and distrust: New relationships and realities*, Academy of Management Review **23** (1998), no. July.
- [56] G. L. Lilien and A. Rangaswamy, *Marketing engineering: Computer-assisted marketing analysis and planning*, Addison-Wesley, Reading, MA, U.S.A., 1998.

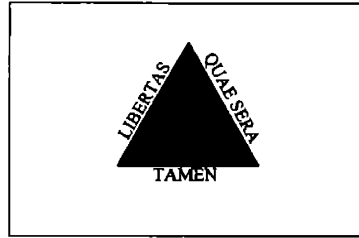
- [57] J.D.C. Little, *Aggregate advertising models: The state of the art*, Operations Research **27** (1979), no. 4, 629–667.
- [58] M.L. Littman, *The witness algorithm: Solving partially observable markov decision processes*, Department of computer science, technical report, Brown University, Providence, R.I., 1994.
- [59] M.L. Littman, A.R. Cassandra, and L.P. Kaelbling, *Learning policies for partially observable environments: Scaling up*, Proceedings of the Twelfth International Conference on Machine Learning (A. Frieditis and S. Russell, eds.), Morgan Kaufmann, San Francisco, CA, 1995, pp. 362–370.
- [60] W.S. Lovejoy, *Some monotonicity results for partially observed markov decision processes*, Operations Research **35** (1987), no. 5, 736–743.
- [61] ———, *Computationally feasible bounds for partially observed markov decision processes*, Operations Research **39** (1991), no. 1, 162–175.
- [62] ———, *A survey of algorithmic methods for partially observed markov decision processes*, Annals of Operations Research **28** (1991), no. 1, 47–66.
- [63] ———, *Suboptimal policies, with bounds, for parameter adaptive decision processes*, Operations Research **41** (1993), no. 3, 583–599.
- [64] P. Maes, *Intelligent software*, Scientific American **273** (1995), no. 9, 84–86.
- [65] R.L. Marcellus and M. Dada, *Interactive process quality improvement*, Management Science **37** (1991), no. 11, 1365–1376.
- [66] D. McFadden, *Econometric models of probabilistic choice*, Structural Analysis of Discrete Data with Econometric Applications; C.F. Mansky and D.McFadden (Eds.).
- [67] Robert J. Meyer and Barbara E. Kahn, *Probabilistic models of consumer choice behavior*, Handbook of Consumer Theory and Research, Eds T. Robertson and H. Kassarian, Prentice-Hall (1990).
- [68] T.M. Mitchell, *Machine learning*, McGraw Hill, New York, NY, U.S.A., 1997.
- [69] A. Mody, *Firm strategies for costly engineering learning*, Management Science **35** (1989), no. 4, 496–512.
- [70] G. Monahan, *A survey of partially observable markov decision processes: Theory, models, and algorithms*, Management Science **28** (1982), no. 1, 1–15.
- [71] R.M. Morgan and S.D. Hunt, *The commitment-trust theory of relationship marketing*, Journal of Marketing **58** (1994), no. July, 20–39.

- [72] Zeithaml V.A. Parasuraman, A. and L. Berry, *A conceptual model of service quality and its implications for future research*, Journal of Marketing **49** (1985), no. Fall, 41–50.
- [73] S.M. Pollock, *A simple model of search for a moving target*, Operations Research **18** (1970), 883–903.
- [74] P. Poupart and C. Boutilier, *Value-directed sampling methods for monitoring pomdps*, Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, Stanford, CA, 2000, pp. 497–506.
- [75] ———, *Vector-space analysis of belief-state approximation for pomdps*, Uncertainty in Artificial Intelligence: Proceedings of the Seventeenth Conference (UAI-2001), Morgan Kaufmann Publishers, San Francisco, CA, 2001, pp. 445–452.
- [76] D. Precup and R.S. Sutton, *Multi-time models for temporally abstract planning*, NIPS-11 (M. Mozer, M. Jordan, and T. Petsche, eds.), MIT Press, Cambridge, MA, 1998.
- [77] ———, *Theoretical results on reinforcement learning with temporally abstract behaviors*, Tenth European Conference on Machine Learning, Chemnitz, Germany, 1998.
- [78] F. Reichheld, *The loyalty effect*, Harvard Business School Press, 1996.
- [79] S. Rosen, *Learning by experience as joint production*, Quarterly Journal of Economics **86** (1972), 366–382.
- [80] D. Rosenfeld, *Markovian deterioration with uncertain information*, Operations Research **24** (1976), no. 1, 141–155.
- [81] S.M. Ross, *Quality control under markovian deterioration*, Management Science **17** (1971), no. 9, 587–596.
- [82] P. Rossi, R. McCulloch, and G. Allenby, *The value of purchase history data in target marketing*, Marketing Science **15** (1996), no. 4, 321–340.
- [83] W. Rudin, *Principles of mathematical analysis*, McGraw-Hill, 1976.
- [84] M.B. Sawaki, *Transformation of partially observable markov decision processes into piecewise linear ones*, Journal of Mathematical Analysis and Applications **91** (1983), 112–118.
- [85] R. Smallwood and E. Sondik, *The optimal control of a partially observable markov decision process over a finite horizon*, Operations Research **21** (1973), 1071–1088.
- [86] E. Sondik, *The optimal control of partially observable markov processes*, Ph.d. dissertation, Stanford University, Palo Alto, CA, 1971.

- [87] E.J. Sondik, *The optimal control of a partially observable markov decision process over the infinite horizon: Discounted costs*, Operations Research **26** (1978), 282–304.
- [88] R.S. Sutton and A.G. Barto, *Reinforcement learning: An introduction*, MIT Press, Cambridge, MA, U.S.A., 1998.
- [89] C. Terwiesch and R.E. Bohn, *Learning and process improvement during production ramp-up*, International Journal of Production Economics **70** (2001), 1–19.
- [90] S. Thomke and D.E. Bell, *Sequential testing in product development*, Management Science **47** (2001), no. 2, 308–323.
- [91] L.L. Thurstone, *A law of comparative judgement*, Psychological Review **34** (1927), 273–286.
- [92] O. Toubia, D.R. Simester, and J.R. Hauser, *Fast polyhedral adaptive conjoint estimation*, Working paper, MIT Sloan School of Management, Cambridge, MA, USA, 2001.
- [93] G.L. Urban, F. Sultan, and W.J. Qualls, *Placing trust at the center of your internet strategy*, Sloan Management Review **42** (2000), no. 1, 39–48.
- [94] K. M. Van Hee, *Bayesian control of markov chains*, Working paper, Mathematical Center Tract 95, Amsterdam, Holland, 1978.
- [95] C.J.C.H. Watkins, *Learning from delayed rewards*, Ph.d. thesis, King’s College, University of Cambridge, Cambridge, U.K., 1989.
- [96] C.J.C.H. Watkins and P. Dayan, *Q-learning*, Machine Learning **8** (1992), no. 3, 279–292.
- [97] C.C. White III, *Applications of two inequality results for concave functions to a stochastic optimization problem*, Journal of Mathematical Analysis and Applications **55** (1976), 347–350.
- [98] ———, *Optimal diagnostic questionnaires which allow less than truthful responses*, Information and Control **32** (1976), 61–74.
- [99] ———, *A markov quality control process subject to partial observation*, Management Science **23** (1977), no. 8, 843–852.
- [100] ———, *A survey of solution techniques for partially observed markov decision processes*, Annals of Operations Research **32** (1991), 215–230.
- [101] C.C. White III and W.R. Scherer, *Solution procedures for partially observed markov decision processes*, Operations Research **37** (1991), no. 5, 791–797.

- [102] M.B. Woodroffe, *A one-armed bandit problem with a concomitant variable*, Journal of the American Statistical Association **74** (1979), 799–806.
- [103] N.L. Zhang and S.S. Lee, *Planning with partially observable markov decision processes: Advances in exact solution method*, Proceedings of the fourteenth conference on uncertainty in artificial intelligence, University of Wisconsin Business School, Madison, WI, USA, 1998.





defined by the inner-product

$$((v, w)) := v(t_f)^T W_T w(t_f) + \int_{t_0}^{t_f} v(t)^T W_R w(t) dt.$$

Unfortunately, we are not yet ready to apply the TRCG algorithm due to a difficulty that has been introduced: system (5.19,5.20) now depends on  $\Delta y_i$ , and is, as a result, no longer decoupled in time. We propose to use a *known approximation* for the nonlinear term  $(K \nabla^2 f(y_i) \lambda_i \Delta y_i)$  in equation (5.20), thus decoupling the system in time and allowing for the use of the TRCG algorithm.

In solving for the inhomogenous terms, we solve the *uncoupled* system

$$\dot{\Delta y}_I = \tilde{A} \Delta y_I - \tilde{Q}_i^k \Delta \lambda_I + \tilde{F}_i, \quad \Delta y_I(0) = y_0 - y_i(t_0), \quad (5.23)$$

$$-\Delta \dot{\lambda}_I = \tilde{A}^T \Delta \lambda_I - K \nabla^2 f(y_i) \lambda_i \tilde{\Delta y}_i + \tilde{\Gamma}_i, \quad \Delta \lambda_I(t_f) = -W_T \tilde{y}_{T,i}, \quad (5.24)$$

where  $\tilde{\Delta y}_i$  is an approximation of  $\Delta y_i$ . Given a method for determining  $\tilde{\Delta y}_i$ , system (5.23,5.24) becomes uncoupled in time, allowing for the calculation of  $\Delta y_I$  and the subsequent use of the TRCG algorithm.

We must therefore finally propose a method for determining  $\tilde{\Delta y}_i$ . We do so iteratively: given an index  $r$  and an iterate  $\tilde{\Delta y}_{i,r}$ , solve equations (5.23,5.24) with

$$\tilde{\Delta y}_i = \tilde{\Delta y}_{i,r}. \quad (5.25)$$

We denote the result of this operation  $\tilde{\Delta y}_{I,r}$ . With this value, we can use the TRCG algorithm to solve

$$\boxed{\mathcal{G} \tilde{\Delta y}_{i,r+1} = \tilde{\Delta y}_{I,r}} \quad (5.26)$$

for the next iterate  $\tilde{\Delta y}_{i,r+1}$ . These iteration can then be repeated until some stopping criterion,  $\|\tilde{\Delta y}_{i,r+1} - \tilde{\Delta y}_{i,r}\| < \epsilon$  for example, is observed. The crucial point is that since  $\mathcal{G}$  is SPD in the inner-product space defined above, the TRCG algorithm can be used without modifications to effectively solve system (5.13) for  $\Delta y_i$ .

### 5.4.3 Initializing the Algorithm

In order to execute the algorithm proposed above, an initial guess must be specified. Here we address the final issue of choosing a suitable  $\tilde{\Delta y}_{i,0}$  ( $r = 0$ ). Suppose we choose  $\tilde{\Delta y}_{i,0} = 0$  to start

the algorithm. Then, solving the system

$$\mathcal{G}\tilde{\Delta y}_{i1} = \tilde{\Delta y}_{I0}$$

for  $\tilde{\Delta y}_{i1}$  provides us with an initial, first order approximation of  $\Delta y_i$ . As we have shown in previous chapters, a solution for the above equation exists, is unique, and can be found by the TRCG algorithm. This is also true of each subsequent iteration ( $r = 1, 2, 3, \dots$ ) regarding the solution of the system (5.26).

Thus every Newton step  $\Delta p_i$  can be calculated assuming the above procedure converges:  $\tilde{\Delta y}_{ir} \rightarrow \Delta y_i$  as  $r \rightarrow \infty$ . Here we make this assumption and show that it holds for our heat radiation example. In addition, we note that the algorithm can only be considered efficient if the iteration converges quickly; that is,  $r$  must not be too large before we attain sufficient convergence.

## 5.5 Sufficient Convergence for Lagging Procedure

It was noted in section 4.4.7 that exact convergence of the Newton iterations for IPMs is not necessary for convergence of the full algorithm. The interpretation in that section was that the solution need not move exactly along the central path for convergence to the true solution  $u^*$ .

Here we are faced with a similar situation. Define the central path of the lagging procedure as the exact solution of the stationary conditions (5.8)-(5.12) for all positive  $\mu^k \in \mathcal{R}$ . Again, we are not interested in the exact value of  $\Delta p_i$  for any given Newton iteration  $i$ , but only in obtaining good enough estimates of these values along the central path as we approach the true solution of the problem. Therefore, it may be postulated that, similarly to IPMs, our lagging procedure need not produce estimates  $\tilde{\Delta p}_i$  that have converged fully to  $\Delta p_i$  of system (5.13).

We further postulate that, much like in the case of IPMs, our estimates can be rather far from the central path and allow for convergence. Here we've applied a very simple rule to take advantage of this feature: take  $r_{\text{last}} = R$  where  $R$  is a small integer, typically less than 5. In section 5.8 we present an example of the fully implemented algorithm where this simple rule has been successfully used.

## 5.6 Determining the Newton Step

Having determined an appropriately “close” approximation  $\tilde{\Delta}y_i$  of  $\Delta y_i$ , we may proceed to find and approximation of the Newton step  $\Delta p_i$ , which we denote  $\tilde{\Delta}p_i$ :

$$\tilde{K}_{PD}\tilde{\Delta}p_i = \tilde{f}_{PD}, \quad (5.27)$$

where now

$$\tilde{K}_{PD} = \begin{bmatrix} W_U & B^T & 0 & -I & I \\ B & 0 & \left(\tilde{A} - \frac{\partial}{\partial t}\right) & 0 & 0 \\ 0 & \left(\tilde{A}^T + \frac{\partial}{\partial t}\right) & W_R & 0 & 0 \\ Z_{1,i} & 0 & 0 & C_{1,i} & 0 \\ Z_{2,i} & 0 & 0 & 0 & C_{1,i} \end{bmatrix},$$

as in section (4.4.8). The above fully determines  $\tilde{\Delta}p_i$  since the right-hand side  $\tilde{f}_{PD}$  now includes the approximation term  $\tilde{\Delta}y_i$ .

## 5.7 NL-IPM-TRCG Algorithm

### Algorithm NL-IPM-TRCG

Set  $k = 0$ ;  
 Set  $\mu^{k=0} = M$  ( $M$  large);  
 Set  $u_{k=0}$  at the analytical center of  $\mathcal{U}$ ;  
 Calculate  $p_{k=0}(u_0)$ ;  
**while** ( $\mu^k > \mu_{\text{tol}}$ ) **do**  
   **while** ( $\|\tilde{F}_i\| > F_{\text{tol}}$ ) **do**  
      $\tilde{\Delta}y_{i_{r=0}} = 0$ ;  
     **for**  $r = 1, \dots, R$  **do**  
        $\tilde{\Delta}y_i = \tilde{\Delta}y_{i_r}$ ;  
       Solve (5.23,5.24) for  $\tilde{\Delta}y_{I_r}$ ;  
       Solve (5.26) by TRCG for  $\tilde{\Delta}y_{i_{r+1}}$ ;  
     **end do**;  
     Solve (5.27) for  $\tilde{\Delta}p_i$ ;

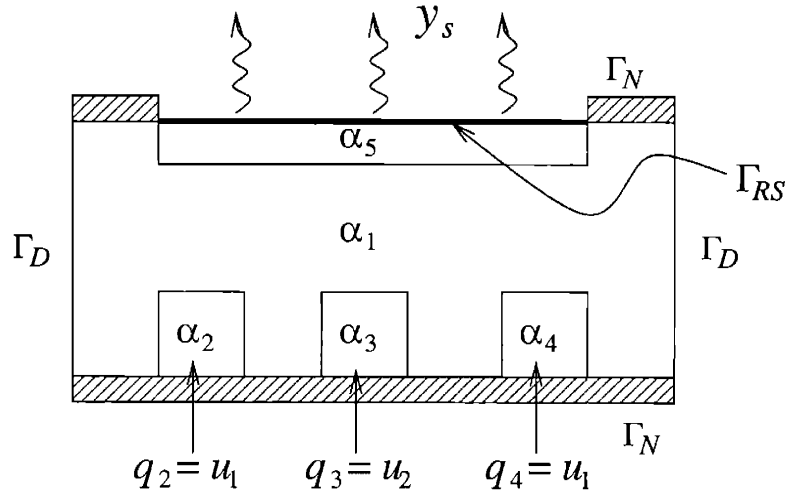


Figure 5-1: Diagram of sample nonlinear heat transfer problem domain (7 cm  $\times$  3 cm).

Solve for  $\alpha_i$  and  $\beta_i$  by Armijo rule and (4.65), respectively;

$$p_{i+1} = p_i + (0.995)\beta_i\alpha_i\tilde{\Delta}p_i;$$

**end do;**

$$p^k = p_i;$$

**if** ( $\mu^k > \mu_{\text{tol}}$ ) **then**

$$k = k + 1;$$

Solve for  $\mu^k$  by (4.64);

**end if;**

**end do**

## 5.8 Example Problem: Nonlinear, Constrained 2D Heat Transfer

### 5.8.1 Problem Statement

We now address a specific nonlinear problem governed by partial differential equations. In particular, we consider radiative heat transfer, where the geometry is similar to the one considered in the problems of previous chapters.

Take the domain shown in Figure 5-1, where the reaction surface  $\Gamma_{RS}$  is now exposed to an environment at temperature  $y_s$ . Rather than imposing a Neumann condition on this boundary, we allow heat exchange through radiation to occur. This process is nonlinear and governed by the Stephan-Boltzmann law (5.1). The governing equations for this process can thus be expressed as:

$$\tilde{y}(t_0) = \tilde{y}_0 \quad \text{in } \Omega, \quad (5.28)$$

$$\frac{\partial \tilde{y}}{\partial t} = \nabla \cdot (\alpha(x) \nabla \tilde{y}) + \sum_{m=1}^M u_m(x) \quad \text{in } \Omega \times (t_0, t_f), \quad (5.29)$$

$$\nabla \tilde{y} \cdot \hat{n} = 0 \quad \text{on } \Gamma_N \times (t_0, t_f), \quad (5.30)$$

$$\nabla \tilde{y} \cdot \hat{n} = \tilde{\sigma}(\tilde{y}^4 - y_s^4) \quad \text{on } \Gamma_{RS} \times (t_0, t_f), \quad (5.31)$$

$$\tilde{y} = 300 \text{ K} \quad \text{on } \Gamma_D \times (t_0, t_f), \quad (5.32)$$

where all definitions and properties are as in previous chapters. We add here that we take  $\tilde{\sigma} = 5 \times 10^{-7} / \text{m}^2 \text{ K}^3$  for the following examples. This value is unrealistically high for typical engineering materials, making the effect of the nonlinear term more pronounced than may be expected to test the proposed algorithm.

### 5.8.2 FEM Formulation

In stating the above problem in the FEM context, we recall the spaces defined in section 3.11. To preserve desirable properties of the stiffness matrix, we deal directly with the time-discretized form of the problem and treat the nonlinearity explicitly. By doing so we may state the problem governing equations as

$$y^0 = y_0; \quad (5.33)$$

$$M \left( \frac{y^\ell - y^{\ell-1}}{\Delta t} \right) = Ay^\ell - f(y^{\ell-1}) + Bu^\ell + F^\ell, \quad \ell = 1, \dots, L, \quad (5.34)$$

where

$$f(y^{\ell-1})_j = (\tilde{\sigma}((\tilde{y}^{\ell-1})^4 - y_s^4), \phi_j). \quad (5.35)$$

Having defined the problem as such, and following the procedures of section 3.11, we may easily derive the stationarity conditions:

$$y^0 = y_0; \quad (5.36)$$

$$M \left( \frac{y^\ell - y^{\ell-1}}{\Delta t} \right) = Ay^\ell - f(y^{\ell-1}) + Bu^\ell + F^\ell, \quad \ell = 1, \dots, L; \quad (5.37)$$

$$(M - \Delta t A)^T \lambda^L = W_T(y^L - y_T) + W_R(y^L - y_R)\Delta t; \quad (5.38)$$

$$M \left( \frac{\lambda^\ell - \lambda^{\ell+1}}{\Delta t} \right) = A^T \lambda^\ell + W_R(y^\ell - y_R^\ell) - \nabla f(y^\ell) \lambda^{\ell+1}, \quad \ell = L-1, \dots, 1; \quad (5.39)$$

$$u^\ell = -W_U^{-1} B^T \lambda^\ell, \quad \ell = 1, \dots, L-1. \quad (5.40)$$

where we define the matrix

$$\nabla f(y^\ell) = 4 \operatorname{diag}[(\bar{\sigma}(\tilde{y}^\ell)^3, \phi)], \quad (5.41)$$

with

$$(v, \phi)_j = (v, \phi_j).$$

Finally, in order to pose the Newton projection problem, we state the variations in stationarity conditions as

$$\Delta y_i^0 = y_0 - y_i^0, \quad (5.42)$$

$$M \left( \frac{\Delta y^\ell - \Delta y^{\ell-1}}{\Delta t} \right) = \tilde{A}_i^\ell \Delta y_i^\ell - \tilde{Q}_i^k \Delta \lambda_i^\ell + \tilde{F}_i^\ell, \quad (5.43)$$

$$(M - \Delta t \tilde{A}_i^L)^T \Delta \lambda^L = W_T(\Delta y^L - \tilde{y}_{T,i}) + W_R(\Delta y^L - \tilde{y}_{R,i}^L)\Delta t; \quad (5.44)$$

$$M \left( \frac{\Delta \lambda^\ell - \Delta \lambda^{\ell+1}}{\Delta t} \right) = (\tilde{A}_i^\ell)^T \Delta \lambda_i^\ell + (W_R - \nabla^2 f(y_i^\ell) \lambda_i^\ell) \Delta y_i^\ell + \tilde{\Gamma}_i^\ell, \quad (5.45)$$

$$\Delta u_i^\ell = -(\tilde{H}_i^k)^{-1} (B^T \lambda_i^\ell + B^T \Delta \lambda_i^\ell + \tilde{g}_i), \quad (5.46)$$

where we note that  $\tilde{F}_i^\ell$  and  $\tilde{\Gamma}_i^\ell$  absorb the explicit nonlinearity terms, and

$$\tilde{A}_i^\ell = A + \nabla f(y_i^\ell), \quad (5.47)$$

$$\nabla^2 f(y_i^\ell) = 12 \operatorname{diag}[(\bar{\sigma}(\tilde{y}_i^\ell)^2, \phi)]. \quad (5.48)$$

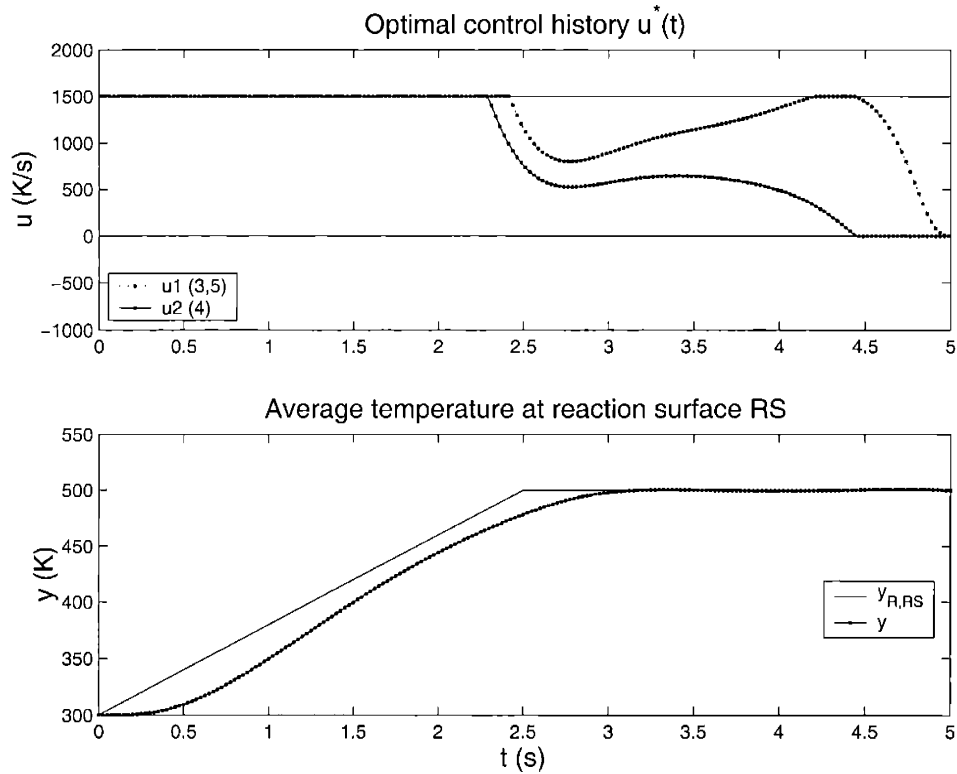


Figure 5-2: Optimal control and  $\Gamma_{RS}$  temperature histories for nonlinear constrained problem,  $J^* = 1.42 \times 10^8$ .

### 5.8.3 General Results

Having thus defined the problem, we may safely apply the NL-IPM-TRCG algorithm to the heat transfer process of Figure 5-1. We preserve all the problem data given in sections 3.12.1 and 4.6.1.

Figure 5-2 presents the solution of optimal control and state histories. We note that the nonlinearity drastically changes the nature of the solution in comparison to the that of section 4.6.1. The heat lost through the radiative surface causes the temperatures at  $\Gamma_{RS}$  to be generally lower throughout the process than was observed with Neumann boundary conditions. In fact, increasing the cost for these deviations will not impact this portion of the solution, since as the controllers saturate in the early part of the process and cannot drive the system to the desired temperature as fast as in the former situation.

We note that if hard bounds were not imposed on the problem, early control values would increase far beyond practical limitations when driven by such strong nonlinearities. This example demonstrates the strength of the TRCG algorithm in that it allows for effective incorporation of both nonlinearities and hard bounds, making it very practical for engineering problems.



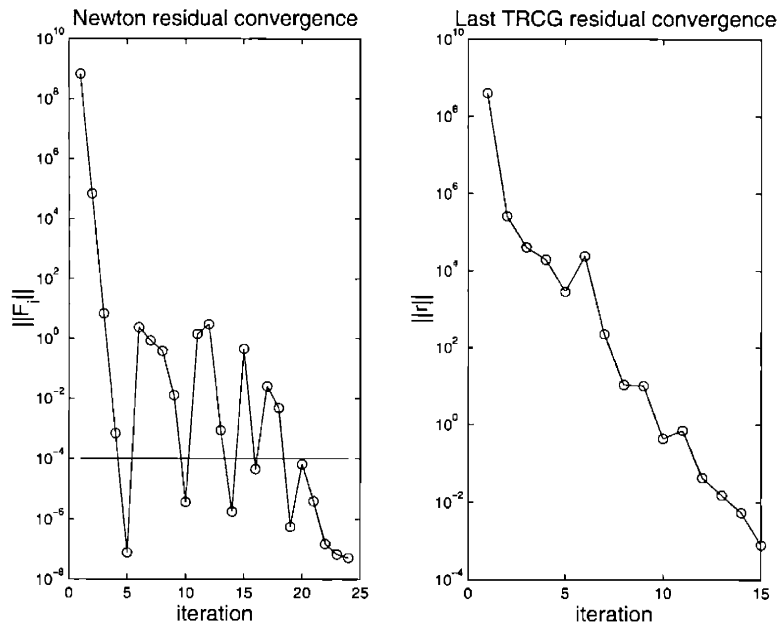


Figure 5-3: Newton and last conjugate gradient convergence of NL-IPM-TRCG algorithm.

#### 5.8.4 Numerical Performance

We chose  $R = 3$  for the NL-IPM-TRCG algorithm, so that each circle shown on the left plot of Figure 5-3 corresponds to 3 iterations to find  $\tilde{\Delta p}_i$ . Comparing Figures 4-2 and 5-3, we conclude that the deviation from the central path introduced by the nonlinearity has a minimal effect on the convergence of Newton iterations when even this small number of lagging iterations is used. New values of barrier parameters  $\mu^k$  were calculated at iterations  $\{1, 6, 11, 15, 17, 20, 21, 22, 23, 24\}$ .

From the right plot of the figure, we note that the last TRCG calculation is still well conditioned, as expected for Primal-Dual methods, converging in only 15 iterations.

The power of the method thus lies in the fact that IPMs tend to be forgiving of deviations from the central path. A small number of lagging iterations (3 or 4) is therefore all that is required to achieve sufficient convergence. The conditioning of the problem will not degrade as  $u^k \rightarrow u^*$  if Primal-Dual IPMs are used, guaranteeing that the TRCG calculations will converge quickly throughout the process. These features, in conjunction with the stability of the central SPD operator  $\mathcal{G}$ , lead to a very efficient overall algorithm.



## Chapter 6

# Concluding Remarks

### 6.1 Summary of Contributions

We presented in this work a method for solving quadratic cost, nonlinear state equations, constrained control optimal control problems. The core of the algorithm was developed for unconstrained LQP, but its flexibility allowed for extensions to more practical problems by applications of IPMs and a lagging technique.

Though developed primarily in the context of ODEs, the goal of this work was to solve problems governed by first-order parabolic partial differential equations. We showed that the method could be very effectively extended to address these problems since the central requirement that  $\mathcal{G}$  be SPD, stable, and well-conditioned was not compromised by a FEM discretization, as would have been the case, for example, if shooting techniques were employed.

We presented an engineering (heat transfer) problem as an example of the efficiency of the method, and showed that its numerical performance was very favorable at all levels: conjugate gradient, Newton projection, and lagging iterations.

The TRCG algorithm is derived from the idea of a state variable operator in the spirit of HUM. Our formulation differs from HUM in that the redefined problem (which can be viewed as a statement of the dual) is minimized over a space defined by a problem-specific inner product. In this space, we showed that the  $\mathcal{G}$  operator is symmetric positive-definite, allowing for the solution of terminal and regulator problems by a conjugate gradient-based method. In addition,  $\mathcal{G}$  is shown to be well-conditioned, thus allowing the method to converge quickly and efficiently.

The most costly part of the algorithm is the *action* of  $\mathcal{G}$ . Therefore, problems that are charac-

terized by dynamical equations with sparse matrices can take advantage of this sparsity. For FEM discretizations of parabolic partial differential equations, an initial value problem can be solved with  $\mathcal{O}(LN)$  operations, where  $N$  and  $L$  are the number of spatial and temporal nodes, respectively. The action of  $\mathcal{G}$  is twice this cost, and, since 20-30 iterations of the conjugate gradient algorithm are required, the entire problem is solved by roughly twice the order of magnitude of a single initial-value problem.

## 6.2 Possible Pitfalls

A certain amount of care must be taken in implementing the TRCG algorithm. Pitfalls, which often arise in the discrete statement of the problem, can be avoided if the following precautions are taken.

There is no way of predicting the correct form of the terminal conditions for  $\lambda$  if the time-discretized form of stationarity conditions is not derived directly from the discretized cost. Though an incorrect terminal condition introduces only  $\mathcal{O}(\Delta t)$  error in the solution, it will likely destroy the SPD property of  $\mathcal{G}$ , possibly compromising conjugate gradient iterations. Therefore, it is recommended that time-discrete stationary conditions be derived from the discretized cost functional, and that the SPD property of  $\mathcal{G}$  be verified by a proof similar to the one found in section 3.8 with an appropriate, discrete inner-product.

The spatial discretization of the problem in the FEM context introduces the mass matrix on the left-hand side of stationary conditions. We observed that as long as we include this matrix in the definition of operator  $\mathcal{R}$ , no modifications need to be done to the algorithm. In fact, any invertible symmetric matrix can be multiplied to the left-hand side of these equations if we define  $\mathcal{R}$  accordingly.

Finally we note that it is best to use Primal-Dual variants of IPMs. This guarantees that  $\mathcal{G}$  is well-conditioned throughout the solution process. Though Primal methods may be easier to implement and may work for some problems, they cannot guarantee this important property of  $\mathcal{G}$  in general.

### 6.3 Conclusions

The examples presented in the work demonstrated the predicted effectiveness of the method. The operator  $\mathcal{G}$  was shown to be central to the efficiency of the algorithm, due to the fact that it is stable, well-conditioned, and SPD in an appropriate inner-product space. In addressing more general problems, we noted that IPMs provided a way of guiding the solution to  $u^*$  without compromising this operator. More general nonlinear problems can very efficiently be addressed in this context, since IPMs allow for deviations from the central path introduced by nonlinearities. In fact, we propose that for unconstrained nonlinear problems, it would be advantageous to apply fictional limits on  $u$  beyond expected values and employ the NL-IPM-TRCG algorithm to exploit this guiding behavior of the method.



## Appendix A

# Additional Time-Discretization Schemes for TRCG

Here we present two additional time-discretization schemes that may be used with TRCG: Crank-Nicholson and Second Order Backward Difference Formulas. The method is shown here to be applicable for these schemes since the main results of chapter 3 hold given appropriate definitions of the  $\mathcal{R}$  operator and  $((\cdot, \cdot))$  inner-product.

As suggested in that chapter, we approach the derivation from the cost functionals. The results in this appendix show that such an approach leads to the appropriate form of operators for general time-discretizations. As a result, TRCG can accommodate any time-discretization scheme provided care is taken in developing  $\mathcal{R}$  and  $((\cdot, \cdot))$ .

For completeness, we present the below in the context of logarithmic barrier functions of chapter 4. This is done to explicitly show the form of augmented cost functionals as new schemes are introduced. Extensions to the lagging procedure for nonlinear problems are trivial.

## A.1 Crank-Nicholson

### A.1.1 Cost Functional Definition

$$\begin{aligned}
\hat{J}^\mu[\hat{u}] &= \frac{1}{2}(\hat{y}^L - \hat{y}_T)^T W_T (\hat{y}^L - \hat{y}_T) + \frac{1}{2} \sum_{\ell=1}^L (\hat{u}^{\ell-1/2})^T W_u (\hat{u}^{\ell-1/2}) \Delta t \\
&+ \frac{1}{2}(\hat{y}^0 - \hat{y}_R^0)^T W_R (\hat{y}^0 - \hat{y}_R^0) \Delta t / 2 + \frac{1}{2} \sum_{\ell=1}^{L-1} (\hat{y}^\ell - \hat{y}_R^\ell)^T W_R (\hat{y}^\ell - \hat{y}_R^\ell) \Delta t \\
&+ \frac{1}{2}(\hat{y}^L - \hat{y}_R^L)^T W_R (\hat{y}^L - \hat{y}_R^L) \Delta t / 2 - \mu \sum_{q=1}^2 \sum_{\ell=1}^L \sum_{m=1}^M \ln(\hat{c}_{m,q}^\ell) \Delta t
\end{aligned} \tag{A.1}$$

where  $\hat{c}_{m,q}^\ell = (\hat{u}_m^{\ell-1/2} - \hat{u}_q^\ell)$ , and  $x^{\ell-1/2}$  is defined as the value of  $x$  at  $t = (\ell - 1/2)\Delta t$ .

### A.1.2 Optimality Conditions

Before applying a particular scheme,  $C_q^\ell$  must be defined. For Crank-Nicholson,  $C_q^\ell = \text{diag}(u_m^{\ell-1/2} - u_q^\ell)$ . We then reintroduce the state and adjoint equations:

$$y^0 = y_0, \tag{A.2}$$

$$\frac{y^\ell - y^{\ell-1}}{\Delta t} = \frac{1}{2}A(y^\ell + y^{\ell-1}) + Bu^{\ell-1/2} + F, \tag{A.3}$$

$$\lambda^L = W_T(y^L - y_T), \tag{A.4}$$

$$\frac{\lambda^\ell - \lambda^{\ell+1}}{\Delta t} = \frac{1}{2}A^T(\lambda^\ell + \lambda^{\ell+1}) + W_R(y^\ell - y_R^\ell), \tag{A.5}$$

$$0 = W_u u^{\ell-1/2} + B^T \lambda^\ell - \mu \left( C_{\min}^{\ell-1} + C_{\max}^{\ell-1} \right) e, \tag{A.6}$$

where equations (A.3-A.6) are used for  $\ell = 1, \dots, L$ .



### A.1.3 TRCG Components

First, the discrete state-adjoint equations are put into linearized form and separated into the inhomogeneous and homogeneous parts:

$$y_I^0 = y_0, \quad (\text{A.7})$$

$$\frac{y_I^{\ell+1} - y_I^\ell}{\Delta t} = \frac{1}{2}A(y_I^{\ell+1} + y_I^\ell) - \frac{1}{2}Q^{\ell+1/2}(\lambda_I^{\ell+1} + \lambda_I^\ell) + D^{\ell+1/2}, \quad (\text{A.8})$$

$$\lambda_I^L = -W_T y_T, \quad (\text{A.9})$$

$$-\frac{\lambda_I^{\ell+1} - \lambda_I^\ell}{\Delta t} = \frac{1}{2}A^T(\lambda_I^{\ell+1} + \lambda_I^\ell) - \frac{1}{2}W_R(y_R^{\ell+1} + y_R^\ell), \quad (\text{A.10})$$

and

$$y_H^0 = 0, \quad (\text{A.11})$$

$$\frac{y_H^{\ell+1} - y_H^\ell}{\Delta t} = \frac{1}{2}A(y_H^{\ell+1} + y_H^\ell) - \frac{1}{2}Q^{\ell+1/2}(\lambda_H^{\ell+1} + \lambda_H^\ell), \quad (\text{A.12})$$

$$\lambda_H^L = W_T y^L, \quad (\text{A.13})$$

$$-\frac{\lambda_H^{\ell+1} - \lambda_H^\ell}{\Delta t} = \frac{1}{2}A^T(\lambda_H^{\ell+1} + \lambda_H^\ell) + \frac{1}{2}W_R(y^{\ell+1} + y^\ell). \quad (\text{A.14})$$

Similarly to the time-continuous case, equations (A.7) - (A.10) are uncoupled, and can be solved for  $y_I^\ell$ , for  $\ell = 1, \dots, L$ . The second set, however, is coupled. Again, we define a RT-operator  $\mathcal{R}_{\text{CN}}$  that solves (A.11) and (A.12) with

$$\lambda_H^L = W_T q^L, \quad (\text{A.15})$$

$$-\frac{\lambda_H^{\ell+1} - \lambda_H^\ell}{\Delta t} = \frac{1}{2}A^T(\lambda_H^{\ell+1} + \lambda_H^\ell) + \frac{1}{2}W_R(q^{\ell+1} + q^\ell), \quad (\text{A.16})$$

such that  $y_H^\ell = \mathcal{R}_{\text{CN}} q^\ell$ ,  $\forall \{q^\ell\}_{\ell=0}^L \in \mathbb{R}^{N \times (L+1)}$ . The problem can then be weakly stated as

$$((p, \mathcal{G}_{\text{CN}} q))_{\text{CN}} = ((p, y_I))_{\text{CN}}, \quad (\text{A.17})$$

where  $\mathcal{G}_{\text{CN}q} = q - \mathcal{R}_{\text{CN}q}$ , and

$$\begin{aligned} ((v, w))_{\text{CN}} &= (v_L)^T W_T(w^\ell) + \frac{1}{2} \sum_{\ell=1}^{L-1} (v^{\ell+1} + v^\ell)^T W_R(w^{\ell+1} + w^\ell) \Delta t, \\ &\forall \{v\}, \{w\} \in \mathbb{R}^{N \times (L+1)}. \end{aligned} \quad (\text{A.18})$$

#### A.1.4 TRCG Proofs

**Proposition 8** *The operator  $((v, w))_{\text{CN}}$  defines an inner-product space.*

*Proof.* Defining the norm  $\|v\|_{\text{CN}} = ((v, v))_{\text{CN}}^{1/2}$ , it is simple to show that for any  $v, w \in \mathbb{R}^{N \times (L+1)}$ :

1.  $((v, w))_{\text{CN}} = ((w, v))_{\text{CN}}$ ;
2.  $\|v\|_{\text{CN}} \geq 0$ ;
3.  $\|v\|_{\text{CN}} = 0 \iff v = 0$ ;
4.  $\|\alpha v\| = |\alpha| \|v\|_{\text{CN}}, \forall \alpha \in \mathbb{R}$ .

□

**Proposition 9** *The R-T operator  $\mathcal{R}_{\text{CN}}$  is symmetric negative semi-definite in the space defined by the above inner product.*

*Proof.* This proof closely follows the discussion for the time-continuous case. We rewrite equations (A.12) and (A.16) in generic variables,  $\{z_1, \gamma_1, \gamma_2, p_1, p_2\} \in \mathbb{R}^{N \times (L+1)}$  with corresponding initial and final conditions:

$$\begin{aligned} \frac{z_1^{\ell+1} - z_1^\ell}{\Delta t} &= \frac{1}{2} A(z_1^{\ell+1} + z_1^\ell) - \frac{1}{2} Q^{\ell+1/2} (\gamma_1^{\ell+1} + \gamma_1^\ell), \quad z_1^0 = 0, \\ -\frac{\gamma_2^{\ell+1} - \gamma_2^\ell}{\Delta t} &= \frac{1}{2} A^T (\gamma_2^{\ell+1} + \gamma_2^\ell) + \frac{1}{2} W_R(p_2^{\ell+1} + p_2^\ell), \quad \gamma_2^L = W_T p_2^L. \end{aligned}$$

Reducing the above set,

$$\begin{aligned} &(\gamma_2^{\ell+1})^T z_1^{\ell+1} - (\gamma_2^\ell)^T z_1^\ell \\ &= -\frac{1}{2} (\gamma_2^{\ell+1} + \gamma_2^\ell)^T Q^{\ell+1/2} (\gamma_1^{\ell+1} + \gamma_1^\ell) \Delta t - \frac{1}{2} (p_2^{\ell+1} + p_2^\ell)^T W_R (z_1^{\ell+1} + z_1^\ell) \Delta t, \end{aligned}$$

and adding from  $\ell = 1$  to  $(L - 1)$ ,

$$\begin{aligned} & (p_2^L)^T W_T z_1^L \\ &= -\frac{1}{2} \sum_{\ell=1}^{L-1} (\gamma_2^{\ell+1} + \gamma_2^\ell)^T Q^{\ell+1/2} (\gamma_1^{\ell+1} + \gamma_1^\ell) \Delta t - \frac{1}{2} \sum_{\ell=1}^{L-1} (p_2^{\ell+1} + p_2^\ell)^T W_R (z_1^{\ell+1} + z_1^\ell) \Delta t, \end{aligned}$$

where we have applied the initial ( $z_1^0 = 0$ ) and final ( $\gamma_2^L = W_T p_2^L$ ) conditions of the problem. Rearranging and applying the operator  $\mathcal{R}_{\text{CN}} p_1^\ell = z_1^\ell$  we obtain

$$\begin{aligned} & (p_2^L)^T W_T \mathcal{R}_{\text{CN}} p_1^L + \frac{1}{2} \sum_{\ell=1}^{L-1} (p_2^{\ell+1} + p_2^\ell)^T W_R (\mathcal{R}_{\text{CN}} p_1^{\ell+1} + \mathcal{R}_{\text{CN}} p_1^\ell) \Delta t \\ &= -\frac{1}{2} \sum_{\ell=1}^{L-1} (\gamma_2^{\ell+1} + \gamma_2^\ell)^T Q^{\ell+1/2} (\gamma_1^{\ell+1} + \gamma_1^\ell) \Delta t. \end{aligned}$$

By identifying the left-hand side of the above equation as  $((p_2, \mathcal{R}_{\text{CN}} p_1))_{\text{CN}}$  we see that

$$((p_2, \mathcal{R}_{\text{CN}} p_1))_{\text{CN}} = ((\mathcal{R}_{\text{CN}} p_1, p_2))_{\text{CN}}$$

and that

$$((p, \mathcal{R}_{\text{CN}} p))_{\text{CN}} = -\frac{1}{2} \sum_{\ell=1}^{L-1} (\gamma_2^{\ell+1} + \gamma_2^\ell)^T Q^{\ell+1/2} (\gamma_1^{\ell+1} + \gamma_1^\ell) \Delta t \leq 0, \quad \forall \{p\} \in \mathbb{R}^{N \times (L+1)}.$$

The operator  $\mathcal{R}_{\text{CN}}$  is therefore SNSD in the  $((\cdot, \cdot))_{\text{CN}}$  space.  $\square$

From the above, we see that  $\mathcal{G}_{\text{CN}}$  ( $\mathcal{G}_{\text{CN}} p = p - \mathcal{R}_{\text{CN}} p$ ) is SPD in  $((\cdot, \cdot))_{\text{CN}}$ , and therefore a unique solution  $q \in \mathbb{R}^{N \times (L+1)}$  of equation (A.17) can be found by conjugate gradient methods in the space defined by this inner product.

## A.2 Second-Order Backward Difference

### A.2.1 Cost Functional Definition

$$\begin{aligned}
\hat{J}^\mu[\hat{u}] &= \frac{1}{2}(\hat{y}^L - \hat{y}_T)^T W_T (\hat{y}^L - \hat{y}_T) + \frac{1}{2}\hat{u}^{1/2} W_u \hat{u}^{1/2} \Delta t \\
&+ \frac{1}{2} \sum_{\ell=2}^L \left( \frac{3}{2}\hat{u}^{\ell-1/2} - \frac{1}{2}\hat{u}^{\ell-3/2} \right)^T W_u \left( \frac{3}{2}\hat{u}^{\ell-1/2} - \frac{1}{2}\hat{u}^{\ell-3/2} \right)^T \Delta t \\
&+ \frac{1}{2}(\hat{y}^0 - \hat{y}_R^0)^T W_R (\hat{y}^0 - \hat{y}_R^0) \Delta t / 2 + \frac{1}{2} \sum_{\ell=1}^{L-1} (\hat{y}^\ell - \hat{y}_R^\ell)^T W_R (\hat{y}^\ell - \hat{y}_R^\ell) \Delta t \\
&+ \frac{1}{2}(\hat{y}^L - \hat{y}_R^L)^T W_R (\hat{y}^L - \hat{y}_R^L) \Delta t / 2 - \mu \sum_{q=1}^2 \sum_{\ell=1}^L \sum_{m=1}^M \ln(\hat{c}_{m,q}^\ell) \Delta t
\end{aligned} \tag{A.19}$$

where  $\hat{c}_{m,q}^\ell = (\hat{u}_m^{1/2} - \hat{u}_q)$  for  $\ell = 1$  and  $\hat{c}_{m,q}^\ell = (\frac{3}{2}\hat{u}_m^{\ell-1/2} - \frac{1}{2}\hat{u}_m^{\ell-3/2} - \hat{u}_q)$  for  $\ell = 2, \dots, L$ .

### A.2.2 Optimality Conditions

For Second-Order BDF,  $C_q^\ell = \text{diag}(u_m^{1/2} - u_q^1)$  for  $\ell = 1$ , and  $C_q^\ell = \text{diag}(\frac{3}{2}u_m^{\ell-1/2} - \frac{1}{2}u_m^{\ell-3/2} - u_q^\ell)$  for  $\ell = 2, \dots, L$ , and

$$y^0 = y_0; \tag{A.20}$$

$$\frac{y^1 - y^0}{\Delta t} = Ay^1 + Bu^{1/2} + F; \tag{A.21}$$

$$\frac{\frac{3}{2}y^\ell - 2y^{\ell-1} + \frac{1}{2}y^{\ell-2}}{\Delta t} = Ay^\ell + B\tilde{u}^\ell + F, \quad \text{for } \ell = 2, \dots, L; \tag{A.22}$$

$$\frac{3}{2}\lambda^L = A^T \lambda^L \Delta t + W_R (y^L - y_R^L) \Delta t / 2 + W_T (y^L - y_T); \tag{A.23}$$

$$\frac{3}{2}\lambda^{L-1} - 2\lambda^L = A^T \lambda^{L-1} \Delta t + W_R (y^{L-1} - y_R^{L-1}); \tag{A.24}$$

$$\frac{\frac{3}{2}\lambda^\ell - 2\lambda^{\ell+1} + \frac{1}{2}\lambda^{\ell+2}}{\Delta t} = A^T \lambda^\ell + W_R (y^\ell - y_R^\ell), \quad \text{for } \ell = L-2, \dots, 2; \tag{A.25}$$

$$\lambda^1 - 2\lambda^2 + \frac{1}{2}\lambda^3 = A^T \lambda^1 \Delta t + W_R (y^1 - y_R^1) \Delta t; \tag{A.26}$$

$$0 = W_u \tilde{u}^\ell + B^T \lambda^\ell - \mu \left( C_{\min}^{\ell-1} + C_{\max}^{\ell-1} \right) e, \quad \text{for } \ell = 1, \dots, L, \tag{A.27}$$

where  $\tilde{u}^\ell = u^{1/2}$  for  $\ell = 1$  and  $\tilde{u}^\ell = \frac{3}{2}u^{\ell-1/2} + \frac{1}{2}u^{\ell-3/2}$  for  $\ell = 2, \dots, L$ .

### A.2.3 TRCG Components

We begin by rewriting the state-adjoint equations:

$$y_I^0 = y_0, \quad \frac{y_I^1 - y_I^0}{\Delta t} = A \left( \frac{2}{3}y_I^1 + \frac{1}{3}y_I^0 \right) - Q^1 \left( \frac{2}{3}\lambda_I^1 \right) + \frac{2}{3}D^1 + \frac{1}{3}D^0, \quad (\text{A.28})$$

$$\frac{\frac{3}{2}y_I^\ell - 2y_I^{\ell-1} + \frac{1}{2}y_I^{\ell-2}}{\Delta t} = Ay_I^\ell - Q^\ell \lambda_I + D^\ell \quad (\text{for } \ell = 2, \dots, L-1), \quad (\text{A.29})$$

$$\lambda_I^{L+1} = 2\lambda_I^L, \quad \lambda_I^L = -W_T y_T, \quad (\text{A.30})$$

$$\frac{\frac{3}{2}\lambda_I^\ell - 2\lambda_I^{\ell+1} + \frac{1}{2}\lambda_I^{\ell+2}}{\Delta t} = A^T \lambda_I^\ell - W_R y_R^\ell \quad (\text{for } \ell = L-1, \dots, 1), \quad (\text{A.31})$$

and

$$y_H^0 = 0, \quad \frac{y_H^1 - y_H^0}{\Delta t} = A \left( \frac{2}{3}y_H^1 + \frac{1}{3}y_H^0 \right) - Q^\ell \left( \frac{2}{3}\lambda_H^1 \right), \quad (\text{A.32})$$

$$\frac{\frac{3}{2}y_H^\ell - 2y_H^{\ell-1} + \frac{1}{2}y_H^{\ell-2}}{\Delta t} = Ay_H^\ell - Q^\ell \lambda_H^\ell \quad (\text{for } \ell = 2, \dots, L-1), \quad (\text{A.33})$$

$$\lambda_H^{L+1} = 2\lambda_H^L, \quad \lambda_H^L = W_T y^L, \quad (y^L = 2y^{L-1} - y^{L-2}) \quad (\text{A.34})$$

$$\frac{\frac{3}{2}\lambda_H^\ell - 2\lambda_H^{\ell+1} + \frac{1}{2}\lambda_H^{\ell+2}}{\Delta t} = A^T \lambda_H^\ell + W_R y^\ell \quad (\text{for } \ell = L-1, \dots, 1), \quad (\text{A.35})$$

where the first set of equations is uncoupled (and can be solved for  $y_I^\ell$ , for  $\ell = 1, \dots, L-1$ , and  $y_I^L = 2y_I^{L-1} - y_I^{L-2}$ ) while the second set is coupled. We define  $\mathcal{R}_{\text{BD}}$  as solving (A.32) and (A.33) with

$$\lambda_H^{L+1} = 2\lambda_H^L, \quad \lambda_H^L = W_T q^L, \quad (q^L = 2q^{L-1} - q^{L-2}) \quad (\text{A.36})$$

$$\frac{\frac{3}{2}\lambda_H^\ell - 2\lambda_H^{\ell+1} + \frac{1}{2}\lambda_H^{\ell+2}}{\Delta t} = A^T \lambda_H^\ell + W_R q^\ell \quad (\text{for } \ell = L-1, \dots, 1), \quad (\text{A.37})$$

such that given  $\{q^\ell\}_{\ell=0}^{L-1} \in \mathbb{R}^{N \times L}$ ,  $(\mathcal{R}_{\text{BD}}q)^\ell = y_H^\ell$  for  $\ell = 1, \dots, L-1$ , and  $(\mathcal{R}_{\text{BD}}q)^L = 2y_H^{L-1} - y_H^{L-2}$ .

The problem can then be weakly stated as

$$((p, \mathcal{G}_{\text{BD}}q))_{\text{BD}} = ((p, y_I))_{\text{BD}}, \quad (\text{A.38})$$

where  $\mathcal{G}_{\text{BD}}q = q - \mathcal{R}_{\text{BD}}q$ , and

$$\begin{aligned} ((v, w))_{\text{BD}} &= \frac{1}{2}(2v^{L-1} - v^{L-2})^T W_T (2w^{L-1} - w^{L-2}) + \sum_{\ell=2}^{L-1} (v^\ell)^T W_R (w^\ell) \Delta t, \\ &\forall \{v, w\}_{\ell=0}^{L-1} \in \mathbb{R}^{N \times L}. \end{aligned} \quad (\text{A.39})$$

#### A.2.4 TRCG Proofs

**Proposition 10** *The operator  $((v, w))_{\text{BD}}$  defines an inner-product space.*

*Proof.* Defining the norm  $\|v\|_{\text{BD}} = ((v, v))_{\text{BD}}^{1/2}$ , it is simple to show that for any  $v, w \in \mathbb{R}^{N \times L}$ :

1.  $((v, w))_{\text{BD}} = ((w, v))_{\text{BD}}$ ;
2.  $\|v\|_{\text{BD}} \geq 0$ ;
3.  $\|v\|_{\text{BD}} = 0 \iff v = 0$ ;
4.  $\|\alpha v\| = |\alpha| \|v\|_{\text{BD}}, \forall \alpha \in \mathbb{R}$ .

□

**Proposition 11** *The R-T operator  $\mathcal{R}_{\text{BD}}$  is symmetric negative semi-definite relative to the above inner product.*

*Proof.* Rewriting equations (A.33) and (A.37) in generic variables,  $\{z_1, \gamma_1, \gamma_2, p_1, p_2\} \in \mathbb{R}^{N \times L}$ ,

$$\begin{aligned} \frac{3}{2}z_1^\ell - 2z_1^{\ell-1} + \frac{1}{2}z_1^{\ell-2} &= Az_1^\ell \Delta t - Q^\ell \gamma_1^\ell \Delta t \\ \frac{3}{2}\gamma_2^\ell - 2\gamma_2^{\ell+1} + \frac{1}{2}\gamma_2^{\ell+2} &= A^T \gamma_2^\ell \Delta t + W_R p_2^\ell \Delta t. \end{aligned}$$

reducing,

$$-2(\gamma_2^{\ell T} z_1^{\ell-1} - \gamma_2^{\ell+1 T} z_1^\ell) + \frac{1}{2}(\gamma_2^{\ell T} z_1^{\ell-2} - \gamma_2^{\ell+2 T} z_1^\ell) = -\gamma_2^{\ell T} Q^\ell \gamma_1^\ell \Delta t - p_2^{\ell T} W_R z_1^\ell \Delta t,$$

and summing from  $\ell = 2$  to  $(L - 1)$ , we obtain

$$\begin{aligned} & (-2\gamma_2^2 + 1/2\gamma_2^3)^T z_1^1 + \gamma_2^{L^T} (2z_1^{L-1} - 1/2z_1^{L-1}) - 1/2\gamma_2^{L+1^T} z_1^{L-1} \\ &= -\sum_{\ell=2}^{L-1} \gamma_2^{\ell^T} Q^\ell \gamma_1^\ell \Delta t - \sum_{\ell=2}^{L-1} p_2^{\ell^T} W_R z_1^\ell \Delta t, \end{aligned}$$

where  $z_1^0 = 0$  has been used. Now we note that according to (A.28) - (A.37):  $\gamma_2^{L+1} = 2\gamma_2^L$ ;  $\gamma_2^L = W_T p_2^L$ ;  $(\mathcal{R}_{\text{BD}} p_1)^L = 2z_1^{L-1} - z_1^{L-2}$ ; and

$$(-2\gamma_2^2 + 1/2\gamma_2^3)^T z_1^1 = \gamma_2^{1^T} (\Delta t A - 3/2I) (\Delta t A - 3/2I)^{-1} Q^1 \gamma_1^1 \Delta t = \gamma_2^{1^T} Q^1 \gamma_1^1 \Delta t.$$

Substituting these terms where appropriate, and rearranging, we obtain:

$$\frac{1}{2} p_2^{L^T} W_T (\mathcal{R}_{\text{BD}} p_1)^L + \sum_{\ell=2}^{L-1} p_2^{\ell^T} W_R (\mathcal{R}_{\text{BD}} p_1)^\ell \Delta t = -\sum_{\ell=1}^{L-1} \gamma_2^{\ell^T} Q^\ell \gamma_1^\ell \Delta t.$$

By identifying the left-hand side of the above equation as  $((p_2, \mathcal{R}_{\text{BD}} p_1))_{\text{BD}}$  we see that

$$((p_2, \mathcal{R}_{\text{BD}} p_1))_{\text{BD}} = ((\mathcal{R}_{\text{BD}} p_1, p_2))_{\text{BD}}$$

and that

$$((p, \mathcal{R}_{\text{BD}} p))_{\text{BD}} = -\sum_{\ell=1}^{L-1} \gamma^{\ell^T} Q^\ell \gamma^\ell \Delta t \leq 0, \quad \forall \{p\} \in \mathbb{R}^{N \times L}.$$

The operator  $\mathcal{R}_{\text{BD}}$  is therefore SNSD in the  $((\cdot, \cdot))_{\text{BD}}$  space.  $\square$

Similarly to before, the operator  $\mathcal{G}_{\text{BD}}$  is SPD in  $((\cdot, \cdot))_{\text{BD}}$ , and a unique solution  $q \in \mathbb{R}^{N \times L}$  of equation (A.38) can be found by conjugate gradient methods in the space defined by this inner product.





# Bibliography

- [1] K. G. Arvanitis, G. Kalogeropoulos, and S. Giotopoulos. Guaranteed stability margins and singular value properties of the discrete-time linear quadratic optimal regulator. *IMA Journal of Mathematical Control and Information*, 18:299–324, 2001.
- [2] Alex Barclay, Philip E. Gill, and J. Ben Rosen. SQP methods and their application to numerical optimal control. Technical Report NA 97-3, Dept of Mathematics, University of California, San Diego, 1997.
- [3] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 2nd edition, 1999.
- [4] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA, 1997.
- [5] J. T. Betts. Survey of numerical methods for trajectory optimization. *AIAA Journal of Guidance, Control, and Dynamics*, 21:193–207, 1998.
- [6] J. T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. Advances in Design and Control. SIAM, Philadelphia, 2001.
- [7] J. V. Breakwell. The optimization of trajectories. *SIAM*, 7:215–247, 1959.
- [8] A. E. Bryson and Y.-C. Ho. *Applied Optimal Control*. Hemisphere Publishing Corporation, revised edition, 1975.
- [9] Y. Cao, M. Gunzburger, and J. Turner. On exact controllability and convergence of optimal controls to exact controls of parabolic equations. *Optimal Control: Theory, Algorithms, and Applications*, pages 67–83, 1998.
- [10] B. Friedland. *Control System Design - An Introduction to State-Space Methods*. McGraw-Hill, 1986.

- [11] O. Ghattas and J.-H. Bark. Optimal control of two- and three- dimensional incompressible navier-stokes flows. *Journal of computational physics*, 136:231–244, 1997.
- [12] Philip E. Gill, Laurent O. Jay, Michael W. Leonard, Linda R. Petzold, and Vivek Sharma. An SQP method for the optimal control of large-scale dynamical systems. *Journal of Computational and Applied Mathematics*, pages 197–213, 2000.
- [13] R. Glowinski and J. L. Lions. Exact and approximate controllability for distributed systems. *Acta Numerica*, pages 159–333, 1995.
- [14] H. Goldberg and F. Trölsch. On a sqp-multigrid technique for nonlinear parabolic boundary control problems. *Optimal Control: Theory, Algorithms, and Applications*, pages 154–177, 1998.
- [15] J.-W. He, R. Glowinski, R. Metcalfe, A. Nordlander, and J. Periaux. Active control of drag optimization for flow past a circular cylinder. *Journal of Computational Physics*, 163:83–117, 2000.
- [16] V. F. Krotov. *Global Methods in Optimal Control*. Monographs and Textbooks in Pure and Applied Mathematics (195). Marcel Dekker, New York, 1996.
- [17] M. K. Kwak and L. Meirovitch. An algorithm for the computation of optimal control gains for second order matrix equations. *Journal of Sound and Vibration*, 161:45–54, 1993.
- [18] J. L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, Berlin, 1971.
- [19] J. L. Lions. Exact controllability, stabilization and perturbations for distributed systems. *SIAM Review*, 30(1):1–68, 1988.
- [20] A. Miele. Method of particular solutions for linear two-point boundary-value problems. *Journal of Optimization Theory and Application*, 2(4), 1968.
- [21] W. Murray. Some aspects of sequential quadratic programming methods. *Large Scale Optimization with Applications, Part II: Optimal Design and Control*, 93:21–35, 1997.
- [22] W.H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in Fortran 77*, volume 1. Cambridge University Press, second edition, 1996.

- [23] S. S. Ravidran. Proper orthogonal decomposition in optimal control. Technical report, Flow Modeling and Control Branch, Fluid Mechanics and Acoustics Division, NASA Langley Research Center, internet, <http://fmcb.larc.nasa.gov/~ravi>.
- [24] R. W. H. Sargent. Optimal control. *Journal of Computational and Applied Mathematics*, 124:361–371, 2000.
- [25] U. Shaked. Guaranteed stability margins for the discrete-time linear quadratic optimal regulator. *IEEE Transactions on Automatic Control*, AC-31(2), 1986.
- [26] V. Sima. *Algorithms for Linear-Quadratic Optimization*. Monographs and Textbooks in Pure and Applied Mathematics (200). Marcel Dekker, New York, 1996.
- [27] Robert F. Stengel. *Optimal Control and Estimation*. Dover Publications, New York, 1993.
- [28] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, Philadelphia, 1997.
- [29] M. H. Wright. Interior methods for constrained optimization. *Acta Numerica*, pages 341–407, 1992.

