On Optimal Search for a Moving Target

A thesis presented for the degree

of Doctor of Philosophy at the

University of Durham.

Benjamin Paul Jordan

Department of Mathematical Sciences,

University of Durham,

DURHAM, DH1 3LE.

April 1997

Abstract

The work of this thesis is concerned with the following problem and its derivatives.

> Consider the problem of searching for a target which moves randomly between n sites . The movement is modelled with an n state Markov chain. One of the sites is searched at each time $t = 1, 2, \ldots$ until the target is found. Associated with each search of site i is an overlook probability α_i and a cost C_i. Our aim is to determine the policy that will find the target with the minimal average cost.

Notably in the two site case we examine the conjecture that if we let p denote the probability that the target is at site 1, an optimal policy can be defined in terms of a threshold probability P^* such that site 1 is searched if and only if $p \geq P^*$. We show this conjecture to be correct (i) for general $C_1 \neq C_2$ when the overlook probabilities α_i are small and (ii) for general α_i and C_i for a large range of transition laws for the movement. We also derive some properties of the optimal policy for the problem on n sites in the no-overlook case and for the case where each site has the same α_i and C_i. We also examine related problems such as ones in which we have the ability to divide available search resources between different regions, and a couple of machine replacement problems.

Preface : Motivating Example

Imagine you are a Coast Guard team leader, and you have just received a distress call from a sinking ship. You know that there is a liferaft somewhere on the (finite) ocean you patrol. Based on tidal charts and metrological information, you can build up some model for the possible motion of that liferaft. At your disposal you have certain resources - a boat and 2 helicopters, all of which can be used to search for this missing raft. You realise that even if you look in the right place , there is a chance that you won't see the liferaft, as it is small in comparison to large waves. Your aim is to find the liferaft as quickly as possible. How best should you allocate your resources in order to achieve this aim?

The above is a motivational example of the work in this thesis and gives an idea of one of the many applications of search theory. We consider only the simplest cases and develop theory which will hopefully be of use in future research in the field.

Acknowledgements

This work was carried out under the supervision of Dr. Iain M. Macphee, without whom life would have been very difficult, and whose curries and atrocious bowling provided many amusing distractions. Moreover, he furnished the original problem to study, without realising, perhaps, the consequences of his actions.My further thanks go go to all my friends in both the Mathematics Department and beyond, for their support and encouragement over the last three years. Thanks also to my parents, brothers and all the Jaegers. Their help (both financial and emotional) has been beyond necessary limits, and has made the passing of time far easier. Above all, my thanks must go to Kristin for her continued attention and devotion to a problem she has never understood. Without her, this work would never have been completed.

This work has been sponsored throughout by the U.K. Engineering and Physical Sciences Research Council, to whom I am greatly indebted.

Statement of Originality

This work was carried out by the author between October 1992 and June 1995. It has not been submitted for any other degree either at Durham or at any other University. No claim of originality is made for the review material in Chapter 1 or the background mathematics in Chapter 2. Chapters 3 and 4 are based on a joint paper between the author and Dr. I.M. MacPhee, although the original motivation and many of the results are the author's own. Chapter 5 presents material not published before.

Contents

List of Figures

Chapter 1

Introduction - A History of Search Theory

Mathematics is the science which uses easy words for hard ideas.

E. Kasner and J. Newman – Mathematics and the Imagination

Throughout history, one thing that has characterised man's existence has been his desire to search for things he does not know about. This search may take forms as diverse as molecular discovery or of interstellar travel. It is surprising perhaps then that the first definition of search theory as a mathematical concept was not until 1942, when the U.S. Navy began work in its Antisubmarine Warfare Operations Research Group. Its major brief was to examine the German submarine threat in the Atlantic, and many of its results are collected in Koopman's *Search and Screening* [7]. Essentially the group examined the problems of how to find submarines, but Koopman also defined the basic probabilistic concepts of search theory - a prior distribution on target location, the idea of a function relating search effort and probability and detection and the intention of maximising probability of detection (or minimising time to detection) subject to search effort constraints. Over the last 50 years these results have remained fundamental to all aspects of optimal search. Moreover, in spite of the changing priorities in the world, search problems remain much the same in definition - a target (or targets) is lost and we have to find it as efficiently as possible under search constraints.

Over the years following Koopman's basic work, search theory developed in many different directions, although initially lack of computing power demanded that the problems themselves be simple. Hence, before 1975, almost all search literature is concerned with the optimal search for a wholly stationary target, with a mobile searcher. Stone [14] highlights this fact by addressing very few moving target problems. His book covers solutions to a wide variety of stationary problems and subsequent extensions have proved difficult. The book by Ahslwede and Wegener [1] builds on Stone's book and discusses more moving target problems. In about 1975, as computer power increased, people began to tackle moving target problems in a new fashion, looking at algorithmic rather than analytic solutions, reflecting the needs of the end-user. This trend has continued until today, with little development in terms of general theory. The type of problem we are interested in is a one-sided search problem in which the target does not respond to the searcher's actions. This problem is obviously simpler than the more game-theoretic two-sided search problem covered in books like that of Gal [4] and is the one more commonly studied. Essentially such problems can be broken down into 2 classes

(i) Optimal search density problems, where the search effort can be infinitely divided and search in one area does not affect search in other areas at other times. This sort of framework is suitable for problems where the searcher moves much more quickly than the target.

(ii) Optimal searcher path problems, where the allocation of search effort at any one time affects the possible future allocations of search effort. This is suitable when the target is at least as fast as the searcher.

Over the years many different types of problem have been raised in search theory, and a great number of papers have been written on them. For a full review, the reader is referred to the excellent survey by Benkoski et al. [3].

In this thesis ,we look at two main areas of search theory

(i) The problem of search for a target which moves around n sites following a Markov chain. In particular we look at $n = 2$ and $n = 3$. At each time we can search any one site.

(ii) The associated extension of problem (i) in which we have the option of dividing search to any degree any searching more than one site concurrently.

Both of these problems fall into the first class of search problems,and both stem from a 1970 paper by Pollock [11]. In this paper, he develops a basic 2-site search problem with perfect detection in discrete time and space and solves it exactly. Since Pollock's paper little has been published on this type of problem, the only papers of note being by Kan [6], in which an n-site problem is discussed, and Nakai [8] where a 3-site problem is examined. However, both of these papers deal with very tightly constrained sets of parameters and offer little general theory of use to this thesis. Their exact relevance is discussed more fully at the appropriate times. The continuous time analogue of the 2-site search problem was solved by Weber [18] in 1989. The other type of problem we look at in this thesis has also been solved in continuous time by Assaf and Sharlin-Bilitzy [2], while the discrete time version has only ever been studied by Nakai [9].

What this clearly shows is that, coming into this research , I found little history on which to base my ideas. As a result, I used computers to simulate optimal search costs and hint at the possible solutions.

People often say mathematics is a difficult subject in which to research. My response is to try to outline the questions facing mathematician when they have a problem they want to solve.

(i) Does this problem have a solution at all?

(ii) What do we think that solution is?

(iii) Can we rigourously prove that our solution is the actual one? (This is the hard part!)

Computers had helped me to achieve the first two parts. The final part you find before you.

The structure of the thesis is as follows. In Chapter 2, we examine the underlying mathematics required to look at search problems in more detail. In Chapter 3, we look at an n-site problem in discrete time. This work is applied and developed in Chapter 4, where we focus on the 2-site case. Chapter 5 offers further problems for examination, and conjectures as yet unprovable results about them.

My own motivation in this research has always been to work on problems which have real relevance , and it is my hope that the results found in this thesis will achieve that aim. Read on, and judge for yourselves. I hope you find what you seek.

Chapter 2

Decision Processes and Negative Programming

When fighting the forces of darkness, one should not wear one's best trousers.

P.G. Wodehouse – The Code of the Woosters

In this chapter, we will examine search problems in the context of underlying mathematical theory, outlining what has been proven about such problems and providing preliminary results which will be used in later chapters. To begin with, let us reformulate the search problem outlined in the Preface in a more general mathematical format.

> We are searching for a target which can be in one of n sites $S_1, S_2, \ldots, S_n$. At each time point $t = 0, 1, \ldots$ we choose a site and search it. Associated with each site S_i is a cost C_i and a probability of overlook α_i. The object itself moves from site to site between search times, choosing its next site according to a Markov chain with transition probability matrix $M = (M_{ij})$. Our aim is to find the object while incurring the minimum expected cost.

Such a problem is an example of a Markov decision process with infinite horizon.

There is a sizeable literature on such problems, and much is known in general about their solution. The results of this chapter are based upon the books of Whittle [20] and Ross [12][13], although notation has been changed for convenience and continuity here. Furthermore, it should be noted that this is not meant to be a comprehensive guide to the subject, but merely highlights the important results which have been employed in the research which follows. Interested readers are referred to either of the above texts for further information.

Within the framework of a Markov decision process, we consider a process which we can observe at times $t = 0, 1, \ldots$ to be in one of a number of possible states, which we can label by the non-negative integers $0, 1, 2, \ldots$. After observing the process and its state, we choose an action from a set $\mathcal{A}$ of all possible actions.

In looking at such a process , we must choose actions according to some *policy*, where a policy is any rule for choosing actions. For example a policy might depend on the history of the process up to that point, or it might be entirely randomised, in that it chooses action A with probability P_A. It is difficult to work with such a general structure however, and in fact for the purposes of this thesis it is only necessary for us to look at a subset of policies.

Definition 2.1

A *Markov policy* is a rule for choosing actions. More formally, it is a function from the state space $\mathcal{X}$ and time to the action space $\pi : \mathcal{X} \times \mathbf{N}_0 \to \mathcal{A}$.

∎

An important subclass of policies is the class of *stationary* policies. These are important because under certain common conditions which we will discuss later, the set of stationary policies contains the set of optimal policies. This result was shown by Strauch [15]. The restriction to stationary policies forms the basis for a huge amount of Dynamic

Programming with relevance not only to the problems which follow, but also to renewal theory and a host of other topics.

Definition 2.2

A policy is *stationary* if the action chosen at time t is dependent only on the state of the process at time t, and not on t itself. More formally it is a function from just the state space $\mathcal{X}$ to the action space $f : \mathcal{X} \to \mathcal{A}$.

∎

If the process is in state i at time t, and our policy tells us to choose action a then two things immediately happen :

(i) We incur a cost $C(i, a)$

(ii) The next state of the system is determined by some transition probabilities $P_{ij}(a)$

So if we denote by X_t the state of the process at time t, then

Definition 2.3

$$P_{ij}(a) = P(X_{t+1} = j | X_0, a_0, X_1, a_1, \ldots, X_t = i, a_t = a)$$

∎

and so costs and transition probabilities are only functions of the last state and last action.

It follows that if a stationary policy π is used then the sequence of states $\{X_t : t = 0, 1, 2, \ldots\}$ forms a Markov chain with transition probabilities $P_{ij} = P_{ij}(\pi(i))$. It is for this reason that the process is called a Markov decision process.

Our aim is to find policies which are in some sense optimal, and to do this we need to determine some optimality criterion. In this way, let us consider the cost incurred up to time s, C_s. We are interested in problems where cost is *separable*, i.e. where:

$$C_s = \sum_{t=0}^{s} \beta_0 \beta_1 \ldots \beta_{t-1} C(X_t, a_t)$$

where $C(X_t, a_t)$ denotes instantaneous cost under action a_t, and the βs are discount factors. These discount factors can be interpreted as estimations of how future costs matter less to us than present ones, and moreover that costs decrease in importance as we get further into the future. It should be noted that in the search example we consider, $\beta_t = 1$, $\forall t$. However, we are also concerned with infinite horizon problems, so we are looking to minimise the infinite horizon cost

$$\mathcal{C} = \sum_{t=0}^{\infty} C(X_t, a_t)$$

Now, this infinite sum can have meaning as

$$\mathcal{C} = \lim_{s \to \infty} C_s$$

i.e. the limit of the s-horizon costs.

There are three standard cases for convergence of this limit. They are:

D The discounted case. Instantaneous costs $C(X_t, a_t)$ are bounded and discounting is uniformly strict. i.e. $|C(X_t, a_t)| \leq \overline{C} < \infty$ and $\beta(t) \leq \delta < 1$ for constants $\overline{C}$, δ.

P The positive case. Instantaneous costs are non-positive.

N The negative case. Instantaneous costs are non-negative.

The apparent paradox in titles comes from the historical definition of $C(X_t, a_t)$ as a reward - hence a positive cost becomes a negative reward. It is clear that the search problem we are interested in lies in the third case - negative programming - as all our costs are non-negative in searching. These costs can be interpreted as a cost in time, or a cost in money for renting search equipment/paying searchers etc.

N.B. We will consider only negative programming cases for the remainder of this chapter. Hence, unless otherwise stated, the results shown are generally good only for negative programming examples.

Definition 2.4

If we follow some policy π, then the expected s-horizon and infinite horizon costs will be

$$V_s(\pi, X_0) = E_\pi(C_s|X_0)$$

and

$$V_\infty(\pi, X_0) = E_\pi(\mathcal{C}|X_0)$$

respectively, which in the case of separable costs become

$$V_s(\pi, i) = E_\pi[\sum_{t=0}^{s} C(X_t, a_t)|X_0 = i]$$

$$V_\infty(\pi, i) = E_\pi[\sum_{t=0}^{\infty} C(X_t, a_t)|X_0 = i]$$

∎

Our aim is to find the optimal cost, or to find the policy which costs us the least to follow over an infinite horizon:

Definition 2.5

The optimal s-horizon and infinite-horizon cost functions V_s and V over the set of Markov policies are defined by

$$\begin{aligned} V_s(i) &= \inf_\pi V_s(\pi, i) \\ V(i) &= \inf_\pi V_\infty(\pi, i) \end{aligned}$$

Moreover as the search problem has Markov structure, we know that V is a function of initial state space X_0 only (proof - see Whittle [20], Chapter 22, Theorem 4.1)

∎

Moreover, we can define an optimal policy in the following manner.

Definition 2.6

Policy π^* is optimal if

$$V_\infty(\pi^*, i) = V(i) \qquad\qquad i \geq 0$$

∎

Of course, $V(i)$ may be infinite.

Moreover, as our search problem has separable costs we have a further important result

Theorem 2.1

If $V_s(\pi, i)$ and $V_\infty(\pi, i)$ are the expected s-horizon and infinite horizon costs under policy π as defined in Definition 2.4, then:

(i) $V_s(\pi, .) \to V_\infty(\pi, .)$ as $s \to \infty$, and $V_s(\pi, .)$ is monotonely increasing in s

(ii) $\inf_\pi V_s(\pi, .) \to V_\infty$ as $s \to \infty$ and $V_\infty \leq V$

Proof

See Whittle [20], Chapter 22, Theorem 2.1. $V_s(\pi, .)$ is monotonely increasing as a trivial result of the fact that we are working in the negative programming case.

∎

Hence we can see that our s-horizon cost under policy π tends to the infinite horizon cost as s tends to ∞, and that the lower envelope of s-horizon cost functions tends to some function V_∞ again as s tends to ∞, which is less than or equal to our optimal infinite horizon cost function, V.

Now, if we define $\mathcal{V}$ to be the class of functions $\phi(x)$ from our state space $\mathcal{X}$ to the real numbers and for any stationary Markov policy h, the operators $L(h)$ and $\mathcal{L}$ from $\mathcal{V}$ to $\mathcal{V}$ by:

$$L(h)\phi(x) = E[C(x_t, a_t) + \phi(x_{t+1})|x_t = x, a_t = h(x)]$$

$$\mathcal{L}\phi(x) = \inf_{a \in \mathcal{A}} E[C(x_t, a_t) + \phi(x_{t+1}) | x_t = x, a_t = a]$$

then we have the result:

Theorem 2.2

The infimal infinite horizon future cost $V(i)$ satisfies

$$V = \mathcal{L}V$$

i.e. V solves the equilibrium optimality equation

$$\phi = \mathcal{L}\phi \tag{2.1}$$

If policy h is optimal then also

$$V = L(h)V$$

Proof

See Whittle [20], Chapter 22, Theorem 4.2

∎

Note that

$$V_s(\pi) = L(\pi_1)L(\pi_2)\ldots L(\pi_s)(0)$$

for any policy π, which simplifies to

$$V_s(\pi) = (L(\pi))^s(0)$$

for a stationary Markov policy. Moreover,

$$V_s = \mathcal{L}^s(0) \quad \text{and} \quad V_\infty = \lim_{s \to \infty} \mathcal{L}^s(0)$$

where 0 indicates the element of $\mathcal{V}$ which is identically zero.

However, the important point is that V need not be the unique solution of (2.1) above. Indeed, a solution ϕ of (2.1) can be the cost function $V_\infty(\pi)$ for some legitimate policy π and yet π need not be the optimal policy.

Example 2.1

Consider the following problem. We have a process which has only one state. At each time point we have to make the decision to either let the process continue which costs us nothing, or to stop the process which costs us 1 unit. We can consider the stationary policies:

f_1 = continue

f_2 = stop

Our optimality equation (2.1) becomes

$$\phi = \min(\phi, 1) \tag{2.2}$$

It is clear that the optimal policy is f_1, with infinite horizon cost $V_{f_1} = 0$. However (2.2) is solved by any $\phi \leq 1$. Notably, it is solved by $\phi = 1 = V_{f_2}$. Hence, V_{f_2} solves $\phi = \mathcal{L}\phi$, but policy f_2 is not optimal. It is worth noting, however, that if continuing costs us any amount greater than 0, then the solution to (2.1) is unique, and policy f_2 is optimal.

∎

Returning then, to our search problem, we want to find conditions under which there is a unique function V which satisfies

$$V(i) = \min_{a\in\mathcal{A}}[C(i,a) + \sum_j P_{ij}(a)V(j)] \tag{2.3}$$

We also want to know when $V_\infty(i) = V(i)$ (we know that $V_\infty(i) \leq V(i)$). First we show that V is the smallest non-negative solution of the optimality equation (2.1).

Proposition 2.1

If the non-negative function $u(i)$ is such that

$$u(i) = \min_{a \in \mathcal{A}}[C(i,a) + \sum_j P_{ij}(a)u(j)] \tag{2.4}$$

then $u(i) \geq V(i)$.

Proof

Let g be a policy which takes minimising actions as determined by equation (2.4). Then,

$$\begin{aligned} C(i, g(i)) + \sum_j P_{ij}(g(i))u(j) &= \min_{a \in \mathcal{A}}[C(i,a) + \sum_j P_{ij}(a)u(j)] \\ &= u(i) \end{aligned} \tag{2.5}$$

Now, let us consider an ordinary decision process problem, with a stop action added to the set of actions, which costs $u(i)$ if used in state i ($u(i) > 0$, $\forall i$). Then (2.5) above tells us that if we are in state i immediate stopping is equivalent to using policy g for one stage and then stopping. Hence:

$$E_g(C_n | X_0 = i) + E_g\left(u(X_n) | X_0 = i\right) = u(i) .$$

As the function u is non-negative, we can see

$$V_n(g,i) = E_g(C_n | X_0 = i) \leq u(i)$$

and, letting n tend to ∞ we find

$$V_\infty(g,i) \leq u(i)$$

Finally, using the fact that by definition, $V(i) \leq V_\infty(g, i)$ for any policy g,

$$V(i) \leq u(i) .$$

∎

Now we show that for search problems the optimality equation has a unique solution.

Definition 2.7

A search problem is a negative case Markov decision problem with a finite action set and a special state $\bar{\imath}$ (corresponding to detection of the target) where there is only one action $\bar{a}$ and $C(\bar{\imath}, \bar{a}) = 0$, $P_{\bar{\imath}\bar{\imath}}(\bar{a}) = 1$.

∎

We now can use the following theorem to show uniqueness.

Theorem 2.3

Suppose a search problem has a set of states I such that $\min_a P_{i\bar{\imath}}(a) \geq \alpha > 0$ for all $i \in I$ and $\sum_{j \in I \cup \{\bar{\imath}\}} P_{ij}(a) = 1$ for all states i and actions a. Then the solution to the optimality equation (2.3) is unique.

Proof

First note that for any solution u to (2.4), $u(\bar{\imath}) = 0$. Fix on some state i and suppose that $V(i) = C(i, a_i) + \sum_j P_{ij}(a_i)V(j)$ for some action a_i. Then

$$\begin{aligned} u(i) - V(i) &\leq C(i, a_i) + \sum_j P_{ij}(a_i)u(j) - C(i, a_i) - \sum_j P_{ij}(a_i)V(j) \\ &= \sum_j P_{ij}(a_i)(u(j) - V(j)) \qquad (2.6) \\ &\leq \sum_{j \neq \bar{\imath}} P_{ij}(a_i) \sup_{j \in I} |u(j) - V(j)| \\ &\leq (1 - \alpha) \sup_{j \in I} |u(j) - V(j)| . \end{aligned}$$

As $u \geq V$ and $\alpha > 0$ this implies $\sup_{i \in I} |u(i) - V(i)| = 0$ and finally, by (2.6), that $u(i) = V(i)$ for all states i.

∎

Thinking about our search problem, we have the result that if we can find a cost function which satisfies the optimality equation then it is the optimal one.

We also want to know that we can use successive approximations to numerically approximate the optimal cost function. Recall that

$$V_\infty(i) = \lim_{n \to \infty} V_n(i)$$

where

$$V_n(i) = \min_{a \in \mathcal{A}} [C(i,a) + \sum_j P_{ij}(a) V_{n-1}(j)] \geq V_{n-1}(i)$$

and $V_0 \equiv 0$.

Theorem 2.4

When the action space $\mathcal{A}$ is finite and costs are all strictly positive then $V_n(i)$ converges monotonely to $V_\infty(i)$ and

$$V_\infty(i) = V(i) \qquad i \geq 0.$$

Proof

As we already know that $V_\infty(i) \leq V(i)$, the proof of the theorem is equivalent to proving that $V(i) \leq V_\infty(i)$. By the above proposition, this is true if V_∞ satisfies the optimality equation $V_\infty(i) = \min_{a \in \mathcal{A}} [C(i,a) + \sum_j P_{ij}(a) V_\infty(j)]$.

$$\begin{aligned} V_\infty(i) &= \lim_{n \to \infty} V_n(i) \\ &= \lim_{n \to \infty} \min_{a \in \mathcal{A}} [C(i,a) + \sum_j P_{ij}(a) V_{n-1}(j)] \\ &= \min_{a \in \mathcal{A}} [C(i,a) + \sum_j P_{ij}(a) \lim_{n \to \infty} V_{n-1}(j)] \\ &= \min_{a \in \mathcal{A}} [C(i,a) + \sum_j P_{ij}(a) V_\infty(j)] \end{aligned}$$

Hence the result is shown.

∎

The importance of this is that we now know that $V(i) = V_\infty(i) = \lim_s \mathcal{L}^s V_0(i)$ which shows us that the optimal policy can be calculated by iterating on V_0 with operator $\mathcal{L}$.

The conclusions of Theorem 2.4 are not true in general. For example, consider the following scenario where the action space is not finite.

Example 2.2

Imagine the problem where the state space consists of the integers $\{0, 1, 2, \ldots\}$. From state 1, we have the choice of moving to any of the state $\{2, 3, \ldots\}$. From state x, $x > 2$ we have to move to state $(x-1)$ and from state 2 we have to move to state 0. There is zero cost associated with all moves, except the movement $2 \to 0$ which has unit cost. Pictorially we have:

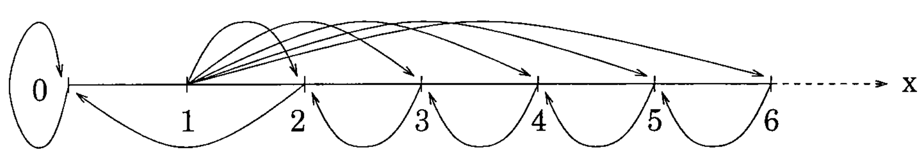


Figure 2.1: Graphical representation of possible transitions in Example 2

We can see easily that, by following a policy which moves us to a state greater than $s+1$ from state 1, we can have

$$V_s(x) = \begin{cases} 0 & \text{if } x = 0, 1 \text{ or } x \geq s+2 \\ 1 & \text{otherwise} \end{cases} \tag{2.7}$$

From (2.8) it follows that

$$V_\infty(x) = \begin{cases} 0 & \text{if } x = 0, 1. \\ 1 & \text{otherwise} \end{cases} \tag{2.8}$$

On the other hand, in an infinite horizon case, no matter where we choose to move from state 1, we ultimately reach state 2, and are forced to make the $2 \to 0$ transition. Hence

$$V(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{otherwise} \end{cases} \tag{2.9}$$

Comparing (2.9) and (2.10), we can see that $V_\infty < V$ at $x = 1$. The proof of Theorem 2.4 fails at the point where the limit and the minimisation are interchanged as here the action space is not finite.

∎

The fact that Theorems 2.3 and 2.4 hold is of particular importance to the work which follows as it forms the justification for all numerical experiments which have been performed. To approximate the optimal policy, we have used value iteration on a discretisation of the state space using Matlab. In this fashion, we have started from $V_0(i) = 0$ and iterated on the operator $\mathcal{L}$ via $V_n = \mathcal{L}V_{n-1}$. Programs for this can be found in the appendices. There are other methods for approximating the optimal solution, notably policy improvement algorithms and linear programming techniques. However, we felt these methods were not appropriate to our particular problem, and so they have not been fully investigated. The concern of this thesis is not to discuss accurate approximations to optimal policies, but rather to prove analytical results about those optimal policies. Naturally, in the course of doing this, we have approximated the optimal policy in order to get some idea about what is happening under certain circumstances, but we felt happy about doing so because we knew that our iterated numerical solutions V_n were always bounded above by the true solution V. We merely used the programs to offer evidence of the true solution, and made them sufficiently accurate to be trusted as such. Further discussion of this can be found in Appendix A. Our aim was never to produce a set of programs which would be particularly quick or elegant. That task is far better left to programmers working closely with the end-users (i.e. the searchers themselves). To this point, however, little has been achieved in this direction using mathematical results. One hope is that results of the sort found in this thesis will be of use in this sort of endeavour. With such thoughts in mind, let us now turn to the search problems themselves.

Chapter 3

The General Search Problem

There's somebody out there for everyone - even if you need a pickaxe, a compass and night goggles to find them.

Harris K. Telemacher (Steve Martin) - L.A. Story

3.1 Introduction

In this chapter, we will look in greater depth at search problems where we seek a moving target on n sites. The main results of this chapter can be found in condensed form in MacPhee and Jordan [5]. They extend those of Nakai [8], and develop the groundwork and basic tools for attacking the specific problems of chapters 4 and 5. To recap, in this chapter, we examine problems of the type:

> We are searching for a target which moves around between n sites choosing its next site according to some Markov chain with transition probability matrix $M = (M_{ij})$. At time points $t = 0, 1, \ldots$, we pick a site and search it. Associated with a search of site i is a cost C_i, and a probability of overlook α_i.

From the results of chapter 2, we know that this is a negative programming problem, and we know methods of approximating the optimal solution. To the author's knowledge, there has been only one paper written on the discrete time search problem on n sites, by Kan [6]. While this paper discusses similar problems to those under consideration here, the aim there is to maximise the probability of detection in a finite amount of time, which is not the same as our aim of minimising cost to detection. Necessarily, the optimal policies can be different. If this were not the case, then the policy which optimised expected cost over infinite time would be the same as the one which optimised probability of detection over unit time - i.e. a myopic one-stage policy. Hence, the results of that paper are of little use to this project.

3.2 Definitions

We denote the current state of the process at time t by the n-vector $p(t) = (p_i(t))$, where $p_i(t) = P(\text{target is in state } i \text{ at time } t)$. If we further denote by S_i the event that the target is in site i, then, at time t, $p_i = P(S_i)$.

At each decision time t, we choose a site and search it. Associated with each site i is an overlook probability α_i, so that even if we look in the right place we may or may not be successful in our endeavour. Let $\mathcal{U}_i$ denote an unsuccessful search of site i. Then

$$\alpha_i = P(\mathcal{U}_i|S_i)$$

The problem is that we don't know where the target is until we find it. What we are trying to do is infer its location from our unsuccessful searches. Using the distribution of the target's position $p(t)$ as our state variable, we can update this after every unsuccessful search using Bayes Theorem.

If we are in state p at time t, and we look in site j and fail to find the target, then the probability that the target is in site i at time $t+1$ is given by

$$P(S_i|\mathcal{U}_j) = \frac{P(S_i \cap \mathcal{U}_j)}{P(\mathcal{U}_j)} \tag{3.1}$$

$$= \frac{\sum_{k \neq j} M_{ki}p_k + \alpha_j M_{ji}p_j}{1-(1-\alpha_j)p_j} \tag{3.2}$$

Hence, following an unsuccessful search of site j, we can update our state variable $p(t)$ using the operator L_j, to $p(t+1)$, where

$$p_i(t+1) = (L_j(p(t)))_i = \frac{\sum_{k \neq j} M_{ki}p_k + \alpha_j M_{ji}p_j}{1-(1-\alpha_j)p_j}. \tag{3.3}$$

For every one-stage action i, we have an operator L_i, which takes us from state to state, and $(L_a(p(t)))_i$ is equivalent to $P_{ij}(a)$ in chapter 2. For simplicity, we will suppress the time dependency notation for the remainder of this chapter, so $p(t)$ will be written p. Let us now turn our attention to the optimality equation.

3.3 The Optimality Equation

We attach a cost C_k to each search of site k. What we want to do is minimise the cost up to detection. Once again, we denote by $\pi(p)$ our search policy i.e. $\pi(p)$ tells us which site to search if we are in state p. Recall that in this case π is dependent only on p, *not* on t, as it is sufficient to merely consider stationary policies as a consequence of Strauch's result [15] (see Defn. 2.2). Let $\mathcal{T}$ denote the time to detection. Then we find (c.f. Definition 2.4)

$$V_\pi(p) = E_\pi(\sum_{t=0}^{\mathcal{T}-1} C_{\pi(p(t))})$$

which we want to minimise by choice of π.

Looking at the search problem, we know that our cost is separable, so we can add the costs of successive searches until the target is found. So

$$V(p;\pi) = C_{\pi(p)} + 0 \times P(\overline{\mathcal{U}}_{\pi(p)}) + V(L_{\pi(p)}(p);\pi)P(\mathcal{U}_{\pi(p)})$$

$$= \quad C_{\pi(p)} + V(L_{\pi(p)}(p);\pi)(1-(1-\alpha_{\pi(p)})p_{\pi(p)})$$

Then, letting $V(p)$ denote $\inf_{\pi} V(p,\pi)$ as before, we find

$$V(p) = \min_k [C_k + V(L_k(p))(1-(1-\alpha_k)p_k)] \tag{3.4}$$

and equation (3.4) is our optimality equation.

Let $\boldsymbol{d} = (d_1, d_2, \ldots)$ $d_i \in \{1, 2, \ldots, n\}$ denote a sequence of search actions, which is chosen independently of p according to some policy, and let $\boldsymbol{d}_{(n)}$ denote the sequence $(d_n, d_{n+1}, \ldots)$. As in chapter 2, we can let $V(p;\boldsymbol{d})$ be the expected cost following actions $\boldsymbol{d}$, starting from state p. Then

$$V(p;\boldsymbol{d}) = C_{d_1} + V(L_{d_1}(p);\boldsymbol{d}_{(2)})P(\mathcal{U}_{d_1}).$$

Now, we can use the fact that $\boldsymbol{d}$ does not depend on $\boldsymbol{p}$ to say

$$V(p;\boldsymbol{d}) = \sum_k p_k V(\boldsymbol{e}_k;\boldsymbol{d}) \tag{3.5}$$

where $\boldsymbol{e}_k$ consists of the probability vector which has all of its mass on site k, which simply says that $V(.,\boldsymbol{d})$ is linear in p. We know from Theorem 2.3 that equation (3.4) has a unique bounded solution. Moreover we can say that

$$V(p) = \inf_{\boldsymbol{d}} V(p;\boldsymbol{d})$$

which says that $V(p)$ is the lower envelope of a family of linear functions and is hence concave.

In addition, we know that we can approximate the optimal cost function $V(p)$ by value iteration using

$$V_{n+1}(p) = \min_k [C_k + V_n(L_k(p))(1-(1-\alpha_k)p_k)]$$

However, our concern is not primarily with the cost function but rather with the optimal policy, as this tells us what we must do in order to find the target in the minimal amount of time. While that amount of time is important, as it gives us some information which can be of use in specific situations (for example, the expected amount of time to find a missing liferaft would be of interest to relatives of the lost people), it is of secondary importance in terms of our work as searchers. We are interested in how to find the target as quickly as possible, independent how long this actually takes, or alternatively, we wish to find the target as cheaply as possible. As our aim is to find the target, the actual cost is not as important as the fact that it is optimal. What we want to know is how to achieve our aim. The way we actually do this is to look at the optimal policy. We know that this is stationary, it tells us that if we are in state p, then it is optimal to perform action a, independent of time. In 1983, Ross [13] conjectured the following intuitively plausible optimal stationary policy in the case where there are $n = 2$ sites.

Ross Conjecture

When we have 2 sites, if we denote p_1 by p, then there is some threshold value P^*, $0 \leq P^* \leq 1$ such that the optimal policy is to search site 2 when $p < P^*$ and to search site 1 when $p > P^*$, with either action being optimal at $p = P^*$.

∎

It is worth noting that this strategy is not the same as the myopic one which tells us to search whichever site gives us the highest chance of immediate success (i.e. if $p_i > p_j \, \forall j \neq i$ then the policy is to search site i), which would be equivalent to $P^* = \frac{1}{2}$ in the 2 site case. To see this consider the following

Example 3.1 *(from Ross [13])*

We seek a target which moves around between 2 sites according to a Markov chain with transition probability matrix

$$M = \begin{pmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Each search of any site costs us 1 unit ($C_1 = C_2 = 1$), and we have perfect detection, so $\alpha_1 = \alpha_2 = 0$.

If $p = 0.55$, searching 1 finds the target with probability 0.55, while searching 2 discovers it with probability 0.45. Hence, a myopic strategy would say that looking in site 1 was a better option. However, an unsuccessful initial search of site 2 leads to certain discovery at the next search (as the target must have been in site 1 before, so will be in site 2 at the next time phase). Unsuccessful search of site 1 results in complete uncertainty as to the whereabouts of the target. Moreover, searching 2 results in expected cost of 1.55 to detection, whereas searching 1 results in *at least* expected cost of $1 + 0.45 \times (1 + 0.5 \times 1) = 1.675$.

∎

3.4 Some Results for the Search Problem on n Sites

When we consider a structure in which there are more than two sites in which the target might be located, our interest is again concerned with the regions $\mathcal{A}_i$ where it is optimal to look initially in site i. The problem is that although Ross' conjecture has a simple and intuitive structure on 2 sites, it has no real analogue in higher dimensions, as it is unclear what is meant by $p > P^*$ in, say, 3 dimensions. Moreover, the value P^* is less important to us, as it doesn't characterise the optimal search regions in the same way as it does in

2 dimensions. In 1973 Nakai [8] showed that in the 3-site case with perfect detection and $C_i = 1$, $\forall i$, the optimal regions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ are *star convex* with respect to the points $(1,0,0)$, $(0,1,0)$, $(0,0,1)$ respectively, where

Definition 3.1

A set S is star convex with respect to the point $p^0 \in S$ if and only if $p \in S$ implies that $\lambda p + (1-\lambda)p^0 \in S$

∎

His argument is quite complicated however, and most of it does not translate to higher dimensions. In the remainder of this chapter, we generalise this result from 3 to $n \geq 2$ sites i.e. we show that the regions $\mathcal{A}_i$, where it is optimal to make the first search in site i, are star convex with respect to the points e_i for $i = 1, \ldots, n$, where e_i is defined to be the probability vector with all its mass on site i, e.g. , in 3 dimensions, $e_1 = (1,0,0)$.

We recall the definitions of the L_i from above, and note that in the no-overlook case $L_i(p) = (1-p_i)^{-1}M^{\mathrm{T}}(p - p_i e_i)$. We can also see $L_i(e_j) = M^{\mathrm{T}}e_j$. $L_i(e_i)$ is not defined in the no-overlook case but this is not important.

It seems to make sense that if there is any state in which it will be optimal to search site i first, it will be the state e_i, as, if we know for certain that the target is in that site, it seems stupid to search anywhere else. Indeed, we can show this is the case under certain conditions by looking at the *subgradients* of the optimal cost function V.

Definition 3.2

For some function $F : \Delta^n \to \mathbf{R}$, with epi$(F) \equiv \{(p,f) : f \leq F(p)\}$, then a hyperplane $(v^T, -1)\begin{pmatrix} p \\ f \end{pmatrix} = k$ (constant) *supports* epi(F) when $(v^T, -1)\begin{pmatrix} p \\ f \end{pmatrix} \geq k$ for all $\begin{pmatrix} p \\ f \end{pmatrix} \in$ epi(F) and $(v^T, -1)\begin{pmatrix} p_0 \\ f_0 \end{pmatrix} = k$ for some $\begin{pmatrix} p_0 \\ f_0 \end{pmatrix} \in$ epi(F) .

A vector v is a subgradient of F when the hyperplane $(v^T, -1)\begin{pmatrix} p \\ f \end{pmatrix} = k$ supports epi(F)

In the context of the search problem we are concerned with, we can say that when a sequence of actions $\boldsymbol{d}$ is optimal at a given p then $(V(e_1\,;\,\boldsymbol{d})\,,\,\ldots,\,,\,V(\boldsymbol{e}_n\,;\,\boldsymbol{d}))$ is a subgradient of $V(\,.\,)$ at p.

∎

Quite a lot can be gleaned from a direct study of these subgradients. When the target is known to be at a particular site we have

$$V(\boldsymbol{e}_i\,;\,(k,\boldsymbol{d})) = \begin{cases} C_i + \alpha_i V(M^{\mathrm{T}}\boldsymbol{e}_i\,;\,\boldsymbol{d})\,, & k = i. \\ C_k + V(M^{\mathrm{T}}\boldsymbol{e}_i\,;\,\boldsymbol{d})\,, & k \neq i. \end{cases} \tag{3.6}$$

Moreover, we can see directly from (3.6) that, if the costs are not site specific (i.e. $C_i = C\,,\,\forall i$), then

$$V\,(\boldsymbol{e}_i\,;\,(i,\boldsymbol{d})) - V\,(\boldsymbol{e}_i\,;\,(k,\boldsymbol{d})) = -(1-\alpha_i)V(M^{\mathrm{T}}\boldsymbol{e}_i\,;\,\boldsymbol{d}) < 0 \tag{3.7}$$

which says that if the target is known to be at site i then it is best to start the search there, as the expected cost will be less. We can now extend Nakai's result to n dimensions.

Theorem 3.1 *(after Nakai)*

In the no-overlook case if $\boldsymbol{e}_i \in \mathcal{A}_i$, the set $\mathcal{A}_i$ of initial probabilities p for which it is optimal to look first in site i is star convex with respect to the point $\boldsymbol{e}_i$.

Proof

Let $V^i(p) = C_i + (1 - \boldsymbol{p}_i)V\,(L_i(p))$. The optimality equation for this case is $V(p) = \min_i V^i(p)$. As $\boldsymbol{e}_i \in \mathcal{A}_i$, $V(\boldsymbol{e}_i) = C_i$. Suppose that in state P it is optimal to first search site i and consider $p = \lambda P + (1-\lambda)\boldsymbol{e}_i$ for $\lambda \in (0,1)$. Now

$$V(p) \;\leq\; V^i(p)$$

$$
\begin{aligned}
&= C_i + (1 - (\lambda P_i + 1 - \lambda))\, V\left((1-p_i)^{-1} M^{\mathrm{T}}(\lambda P + (1-\lambda)e_i - (\lambda P_i + 1 - \lambda)e_i)\right) \\
&= C_i + \lambda(1 - P_i) V\left([\lambda(1-P_i)]^{-1} \lambda M^{\mathrm{T}}(P - P_i e_i)\right) \\
&= \lambda V^i(P) + (1-\lambda) C_i \\
&= \lambda V(P) + (1-\lambda) V(e_i) \le V(p)
\end{aligned}
$$

where the last inequality follows from the concavity of V. Thus it is optimal to search site i first for all $p = \lambda P + (1-\lambda)e_i$ for $\lambda \in (0,1)$ which is precisely what it means for $\mathcal{A}_i$ to be star convex.

∎

Corollary 3.1

In the no-overlook case, with equal costs, the set $\mathcal{A}_i$ of initial probabilities p for which it is optimal to look first in site i is star convex with respect to the point e_i.

Proof

The result comes straight from applying (3.7) to Theorem 3.1.

∎

The case $n = 2$ establishes Ross' conjecture in the no-overlook case, a result which was shown by Pollock in 1970 [11]. Numerical investigation of the case with three sites suggests that the $\mathcal{A}_i$ are not actually convex in general, but rather star convex of the form shown in Figure 3.1 below.

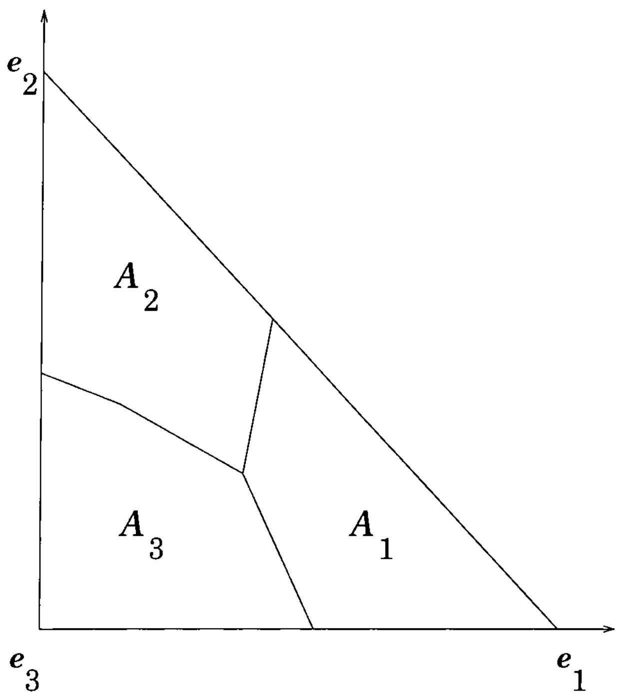


Figure 3.1: Optimal first search regions in three dimensions

When costs are not equal across all sites, it is less clear what we can prove. It is clear from (3.6) that there will be occasions when it is not optimal to look in site i first, even when in state e_i i.e. there will be occasions when $V(e_i) < C_i$. Consider the following example.

Example 3.2

Consider a search problem on 2 sites, with transition probability matrix

$$M = \begin{pmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

We assume no overlook probability, and costs are $C_1 = 3$ and $C_2 = 1$. Then we can clearly see that it is never optimal to look in site 1, so $V(e_1) = 2 \neq C_1$.

∎

Essentially, what we want to be able to say is that, if there is any p such that $V(p) =$

$V_\sigma(p)$ for some policy σ where $\sigma_1 = i$ (i.e. $p \in \mathcal{A}_i$), then $e_i \in \mathcal{A}_i$. However, this conjecture, although seemingly trivial is, in fact probably not always true, as is suggested by the following example, proposed by Penrose [10].

Example 3.3

Consider the 6-state problem with zero overlook probabilities and transition structure as shown in figure 3.2 below.

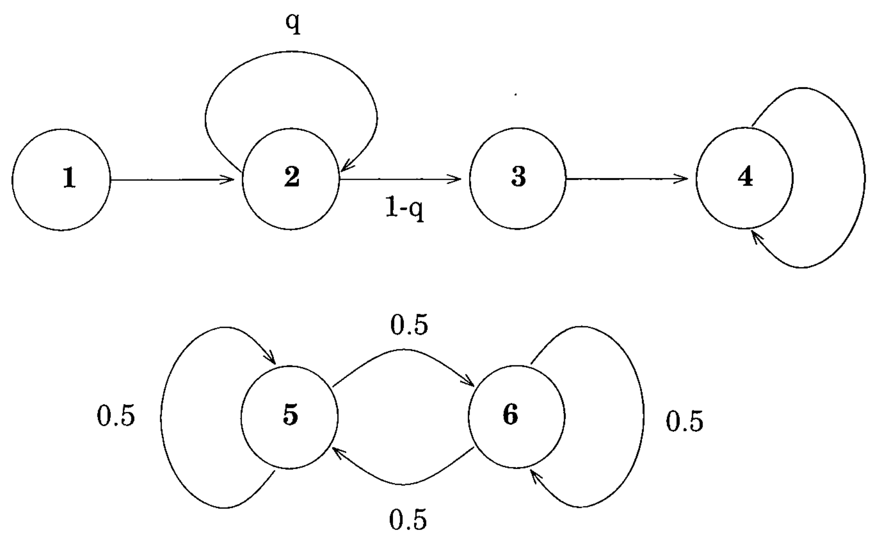


Figure 3.2: Transition structure in Example 3.3

For this example, let us consider costs of $(10, 10^6, 1, 10^6, 5, 10^6)$. Then, when $p_0 = e_1$, strategy {*always look in site 3*} has expected cost $1+\frac{1}{1-q} < 10$ for $q < \frac{8}{9}$ and this strategy must be optimal. When $p_0 = (1-\epsilon)e_1 + 0.5\epsilon e_5 + 0.5\epsilon e_6$ then $V(p_0; 1, 5, 5, 5, \ldots) = 10 + 10\epsilon = 10(1+\epsilon)$.

If we let 5^+ denote the action of searching both site 5 and site 3 simultaneously with cost 5, then

$$\begin{aligned} V(p_0; 5^+, 5^+, 5^+, \ldots) &= 5((1-\epsilon)(1+\frac{1}{1-q}) + 2\epsilon)) \\ &= 5(1+\frac{1}{1-q}) - 5\epsilon\frac{q}{1-q} \end{aligned}$$

and for smallish ϵ the strategy $\{$*always do* $5^+\}$ is worse than strategy $\{$*look in* 1 *once, then always look in* 5$\}$. Of course, the action 5^+ is not actually a legal action, but it may help provide lower bounds for strategies which involve repeated searches of 5, and for this reason it is worth examining.

So now we consider strategy δ_m=$\{$*look in* 3 *m times, then always do* $5^+\}$.

$$\begin{aligned} V(p_0;\delta_m) &= (1-\epsilon)(1+\sum_{j=0}^{m-2} q^j) + \epsilon m + 5((1-\epsilon)q^{m-1}\frac{1}{1-q} + 2\epsilon) \\ &= (1-\epsilon)(1+\frac{1}{1-q} + \frac{4q^{m-1}}{1-q}) + (m+10)\epsilon \end{aligned}$$

Now let $z_m = 4\frac{1-\epsilon}{1-q}q^{m-1} + \epsilon m$. We can see that z_m decreases until $q^m < \frac{\epsilon}{4(1-\epsilon)}$. So let us look at $q = 0.88 < \frac{8}{9}$ and $\epsilon = 0.05$. Then $m > \frac{\log\epsilon - \log 4(1-\epsilon)}{\log q} = 33.88$, so let us take $m^* = 34$ as that is the smallest whole number satisfying the constraints. Then $V(p_0;\delta_{34}) = 9.33 + 2.2 = 11.53$, while $V(p_0;1,5,5,5,5,\ldots) = 10.5$. Obviously there are many other strategies which could be employed, but these seem to be some of the most logical ones, and suggest that a counter-example does exist.

∎

We can say the following, however, which is of particular interest when there are only two sites.

Lemma 3.1

For any pair of sites $i \neq j$ and any action sequence $\boldsymbol{d}$, if $\alpha_i < 1$ and $\alpha_i \leq \min_k \left\{\frac{M_{jk}}{M_{ik}}\right\}$ then $V(\boldsymbol{e}_i\,;\,(i,\boldsymbol{d})) < V(\boldsymbol{e}_j\,;\,(i,\boldsymbol{d}))$.

Proof

For $i \neq j$,

$$\begin{aligned} V(\boldsymbol{e}_i\,;\,(i,\boldsymbol{d})) - V(\boldsymbol{e}_j\,;\,(i,\boldsymbol{d})) &= C_i + \alpha_i V(M^{\mathrm{T}}\boldsymbol{e}_i\,;\,\boldsymbol{d}) - C_i - V(M^{\mathrm{T}}\boldsymbol{e}_j\,;\,\boldsymbol{d}) \\ &= \alpha_i \sum_k M_{ik} V(\boldsymbol{e}_k\,;\,\boldsymbol{d}) - \sum_k M_{jk} V(\boldsymbol{e}_k\,;\,\boldsymbol{d}) \qquad \text{(by (3.6))} \end{aligned}$$

$$= \sum_k V(e_k\,;\, d)(\alpha_i M_{ik} - M_{jk}) \ .$$

The lemma now follows from the inequality on α_i and the observation that $V(p\,;\, d) > 0$ for any proper probability vector p and sequence d since the search costs C_i are positive.

∎

The lemma simply says that a given search sequence will find the target for least cost when the starting position of the target is the same as the initial search site as long as the overlook probabilities are small enough. Unfortunately, this does not mean that we can extend Theorem 3.1 to unequal costs (this would be equivalent to showing that $e_i \in \mathcal{A}_i$, or $V(e_i\,;\,(i,d)) \leq V(e_i\,;\,(j,d))\,,\ \forall j \neq i$). However, we do obtain from this lemma a simple characterisation of $\mathcal{A}_i$ when the α_i are small enough. Suppose that $p \in \mathcal{A}_i$ and that d is an optimal sequence of actions at p. Then $(V(e_j\,;\, d))_j$ is a subgradient to $V(\,.\,)$ so that $\mathcal{A}_i$ consists of those points where the i^{th} component of the subgradient $(V(e_j\,;\, d))_j$ is the minimum component. What does this mean? If we consider $V(p; d)$ as in (3.5), then we can see that

$$V(p\,;\, d) = \sum_{j=1}^{n} p_j V(e_j\,;\, d) = p\hat{V}_d \tag{3.8}$$

where $\hat{V}_d = (V(e_1\,;\, d), \ldots, V(e_n\,;\, d))$. Then $\mathcal{A}_i$ consists of those points where $V\,(e_i\,;\, d) < V\,(e_j\,;\, d) \quad \forall j \neq i$. This is particularly useful in the case of $n = 2$ sites, as it means that $\mathcal{A}_1$ consists of points where action d is optimal and $V(e_1\,;\, d) < V(e_2\,;\, d)$.

<u>**Corollary 3.2**</u>

When there are $n = 2$ sites and $\alpha_1 \leq \min\left\{\frac{M_{21}}{M_{11}}, \frac{M_{22}}{M_{12}}\right\}$, $\alpha_2 \leq \min\left\{\frac{M_{11}}{M_{21}}, \frac{M_{12}}{M_{22}}\right\}$ and $\alpha_1 < 1$, $\alpha_2 < 1$ then Ross' conjecture is correct.

Proof

We will change variable to the scalar p_1 for simplicity. It suffices to show that $V(1; \boldsymbol{d}) - V(0; \boldsymbol{d})$ is negative or positive as $d_1 = 1$ or $d_1 = 2$ respectively. To see this, suppose that a sequence of actions $\boldsymbol{d}$ with $d_1 = 2$ is optimal at $p_1 = \bar{p}$ while a sequence $\hat{\boldsymbol{d}}$ with $\hat{d}_1 = 1$ is optimal at $p_1 = \hat{p} < \bar{p}$. As the linear functions $V(\,.\,; \boldsymbol{d})$ and $V(\,.\,; \hat{\boldsymbol{d}})$ are subgradients of the concave function $V(\,.\,)$ it follows that $V(\bar{p}; \hat{\boldsymbol{d}}) < V(\hat{p}; \hat{\boldsymbol{d}}) \leq V(\hat{p}; \boldsymbol{d}) < V(\bar{p}; \boldsymbol{d})$ but this contradicts the optimality of $\boldsymbol{d}$ at $\bar{p}$. Hence the above conditions force the optimal policy to have threshold form.

That the slope $V(1; \boldsymbol{d}) - V(0; \boldsymbol{d})$ is negative or positive as $d_1 = 1$ or $d_1 = 2$ respectively follows directly from Lemma 3.1 which establishes Ross's conjecture under the stated conditions on the α_i.

∎

In this chapter, then we have examined the general search problem on n sites, and shown that under the assumptions of no overlook and constant search costs, the optimal first-search regions $\mathcal{A}_i$ are star convex with respect to the points e_i. More importantly, in the case of 2 sites, we have shown that Ross' Conjecture is true for a number of different cases, notably when $\alpha_1 \leq \min\left\{\frac{M_{21}}{M_{11}}, \frac{M_{22}}{M_{12}}\right\}$ and $\alpha_2 \leq \min\left\{\frac{M_{11}}{M_{21}}, \frac{M_{12}}{M_{22}}\right\}$ with no restrictions on the values of the C_i. However, we have no idea what the value of P^* actually is in any of these cases, merely the knowledge that such a threshold value exists, and moreover the knowledge that the policy dictated by that threshold value is optimal. With these results in mind, we can now turn our attention to the calculation of exact optimal policies and an extension of Ross' Conjecture to any value of α_i and C_i in two dimensions.

Chapter 4

The 2-Site Problem

One of the principal objects of theoretical research in my department of knowledge is to find the point of view from which the subject appears in its greatest simplicity.
Josiah Willard Gibbs

4.1 Introduction

We now turn our attention to the simplest search problem for a moving target - the search for a target which moves between 2 sites with Markov motion with a known transition probability matrix. In this chapter, we consider such problems with site dependent overlook probabilities, and site dependent search costs (i.e., $\alpha_1 \neq \alpha_2$ and $C_1 \neq C_2$). We know from the results of chapter 3 that under certain conditions, Ross' Conjecture holds in such cases; we seek to give evidence that it is true for all parameter values, and moreover show that in this 2-site case the optimal cost function V is piecewise linear in p, and where possible we seek to give a value to P^*, the threshold parameter suggested by Ross' conjecture. Let us simplify the notation for this case.

4.2 Definitions

Let us say that the target moves between the sites following the transition probability matrix M, where

$$M = \begin{pmatrix} a & 1-a \\ 1-b & b \end{pmatrix} \tag{4.1}$$

i.e. $M_{11} = a$ etc. Let us class our state as being $p = P(\mathcal{S}_1)$. Then $P(\mathcal{S}_2) = 1-p$. Recall the definitions of the functions L_i from (3.3) before. These provide

$$L_1(p) = \frac{\alpha_1 ap + (1-b)(1-p)}{\alpha_1 p + (1-p)} \tag{4.2}$$

and

$$L_2(p) = \frac{ap + \alpha_2(1-b)(1-p)}{p + \alpha_2(1-p)} \tag{4.3}$$

and our optimality equation (3.4) becomes

$$V(p) = \min\{C_1 + V(L_1(p))(\alpha_1 p + (1-p))\,;\, C_2 + V(L_2(p))(p + \alpha_2(1-p))\} \tag{4.4}$$

We conjecture that this cost function is piecewise linear in p. The following is helpful in seeing why this result might be true.

<u>**Example 4.1**</u>

Suppose that $V(t) = A_1 t + B_1$ for t near $L_1(p)$ and $V(t) = A_2 t + B_2$ near $L_2(p)$. Then, by substitution into (4.4) above, we find

$$\begin{aligned} V(p) &= \min\{C_1 + V(L_1(p))(\alpha_1 p + (1-p))\,;\, C_2 + V(L_2(p))(p + \alpha_2(1-p))\} \\ &= \min\{C_1 + (A_1 L_1(p) + B_1)(\alpha_1 p + (1-p))\,;\, C_2 + (A_2 L_2(p) + B_2)(p + \alpha_2(1-p))\} \\ &= \min\{C_1 + A_1(1-b) + B_1 + (A_1\alpha_1 a - A_1(1-b) + B_1\alpha_1 - B_1)p\,; \\ &\qquad C_2 + A_2(1-b)\alpha_2 + B_2\alpha_2 + (A_2 a - A_2(1-b)\alpha_2 + B_2 - B_2\alpha_2)p\} \end{aligned}$$

i.e. we find that V is the minimum of two things that are linear in p.

∎

What this means is that if V is linear on 2 intervals of p, then it will be linear on at least one other. Moreover, if V has a *corner* at some point t (i.e. two separate linear pieces join at t), then there might be another corner at $L_i^{-1}(t)$ where i is determined by whichever of the two expressions in the minimisation (4.4) is less.

One way to interpret this result is to say that V is made up of a certain number of strategies, each of which is optimal on an interval, as graphically shown below

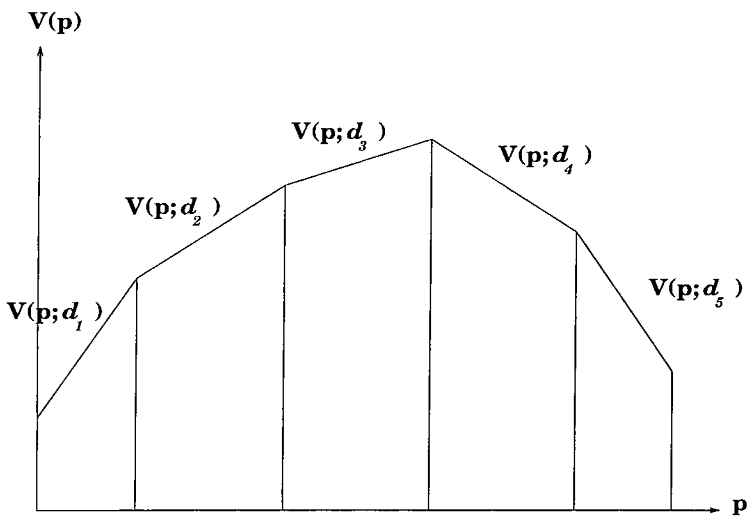


Figure 4.1: Optimal strategies over V

In order to show that V is piecewise linear, we construct such a function, and show that it satisfies (4.4) above. To do this, we first need to look a bit further at $L_1(p)$ and $L_2(p)$.

4.3 Properties of the $L_i(p)$

$L_1(p)$ and $L_2(p)$ are both what is known as *fractional linear mappings*, which are often used as generating functions in branching processes. It is known that they are closed under the composition of functions. To see this, consider the following:

Example 4.2

Let $f(p) = \dfrac{sp+t}{up+v}$ and $F(p) = \dfrac{Sp+T}{Up+V}$ be fractional linear mappings. Then:

$$\begin{aligned} f \circ F(p) &= \frac{s\left(\frac{Sp+T}{Up+V}\right)+t}{u\left(\frac{Sp+T}{Up+V}\right)+v} \\ &= \frac{s(Sp+T)+t(Up+V)}{u(Sp+T)+v(Up+V)} \\ &= \frac{(sS+tU)p+(sT+tV)}{(uS+vU)p+(uT+vV)} \end{aligned}$$

so the composition of two fractional linear mappings is a fractional linear mapping itself.

∎

We are interested only in such mappings f where $v > t > 0$ as $L_i(0) = (1-b) \in [0,1]$ and $f(0) = \dfrac{t}{v}$ and where f is continuous on $(0,1)$. The fractional linear mappings we are concerned with, $L_1(p)$ and $L_2(p)$ can have two representations, so we choose $v > 0$ to pick one of them. Further, we know that $u+v > s+t > 0$, from continuity of f. Note that this also ensures that $up+v \in [0,1]$, $\forall p \in [0,1]$. Let $\mathcal{D} = sv - ut$. We can use this to classify results for these mappings.

Lemma 4.1

Let $f(p) = \dfrac{sp+t}{up+v}$, $p \in [0,1]$ be any linear fractional mapping and suppose that $v > t > 0$ and $u+v > s+t > 0$. Then the equation $f(p) = p$ has a unique solution $q \in [0,1]$, $f(p) > (<)p$ when $p < (>)q$ and further

(i) if $\mathcal{D} = 0$, then $f(p) = \frac{t}{v}$, $\forall p \in [0,1]$;

(ii) if $\mathcal{D} > 0$, then $f(p)$ is increasing and convex when $u < 0$, concave when $u > 0$;

(iii) if $\mathcal{D} < 0$, then $f(p)$ is decreasing and concave when $u < 0$, convex when $u > 0$.

Proof

Consider the itemised cases first. The argument is based on differentiation of f. Note that:

$$f'(p) = \frac{\mathcal{D}}{(up+v)^2} \text{ and } f''(p) = -\frac{2u\mathcal{D}}{(up+v)^3}$$

so these three claims are obvious. Further, the assumptions imply that $0 < f(0), f(1) < 1$ so, by the strict convexity or concavity of f we know there is a unique root q of $f(q) = q$ with $q \in (0,1)$.

It is of interest to actually calculate the value of q as it will be used elsewhere. When $u = 0$, $f(p) = \dfrac{sp+t}{v}$ is linear in p, so the claim $f(p) = p$ has a unique solution is clearly true, with $q = \dfrac{t}{v-s}$. When $u \neq 0$ we can see that as $up+v > 0$ for $p \in [0,1]$, $f(p) = p$ is equivalent to $up^2 + (v-s)p - t = 0$. By analysis of the discriminant of this quadratic we can show there are two real roots to this equation.

$$q = \frac{s-v}{2u} \pm \frac{1}{2u}\sqrt{(v-s)^2 + 4tu}.$$

When $\mathcal{D} > 0$ and $u > 0$ we can see (figure 4.2 below) that $f'(p) \to 0$ as $p \to \infty$, and $f'(p) \to \infty$ as $p \to -\frac{v}{u} < 0$, so the root which is in $(0,1)$ must be the larger root.

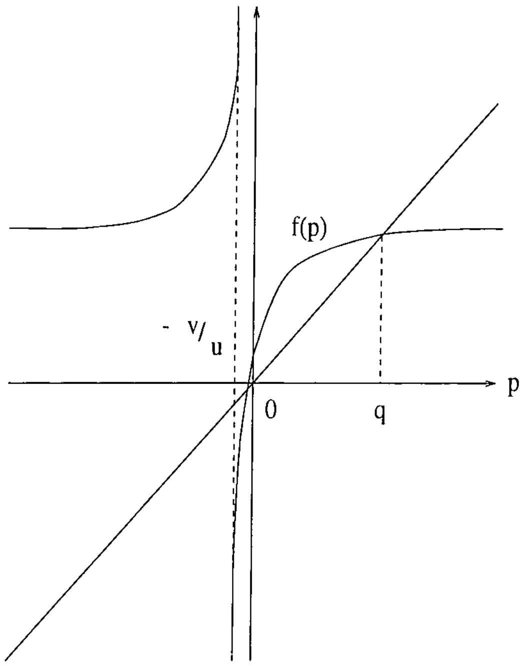


Figure 4.2: Form of $f(p)$ when $\mathcal{D} > 0$ and $u > 0$

When $\mathcal{D} > 0$ and $u < 0$ then $f'(p) \to \infty$ as $p \to -\frac{v}{u} > 1$, so the root in $(0,1)$ is the smaller root. When $\mathcal{D} < 0$ and $u > 0$ then $f'(p) \to 0$ as $p \to \infty$ and $f'(p) \to -\infty$ as $p \to -\frac{v}{u} < 0$, so the root in $(0,1)$ is again the larger root. When $\mathcal{D} < 0$ and $u < 0$ then $f'(p) \to -\infty$ as $p \to -\frac{v}{u} > 1$ and $f'(p) \to 0$ as $p \to -\infty$, and hence the root in $(0,1)$ is the smaller root.

Hence by applying the above analysis to the formulae for q we find that

$$q = \frac{s-v}{2u} + \frac{1}{2u}\sqrt{(v-s)^2 + 4tu}.$$

That $f(p) > p$ for $p < q$ follows immediately from the continuity of f and the fact that $f(0) > 0$. The other inequality is similar.

∎

There are other properties of linear fractional mappings which are of importance to this thesis.

Lemma 4.2

Let f denote a linear fractional mapping with $v > t > 0$ and $u + v > s + t > 0$. Then f is a contraction mapping on an interval around q and when $\mathcal{D} \neq 0$, f^{-1} exists and is increasing or decreasing when f is. Further when

(i) $\mathcal{D} > 0$ then for $p < q$, $f^{-1}(p) < p < f(p) < q$ and $f^{(n)}(p) \to q$ as $n \to \infty$ and is increasing, while for $p > q$, $q < f(p) < p < f^{-1}(p)$ and $f^{(n)}(p) \to q$ as $n \to \infty$ and is decreasing.

(ii) $\mathcal{D} < 0$ then for $p < q$, $q < f(p) < f^{-1}(p)$ while for $p > q$, $f^{-1}(p) < f(p) < q$. For all $p \in [0, 1]$, $| f^{(n)}(p) - q | \to 0$ monotonically as $n \to \infty$ and the differences $f^{(n)}(p) - q$ alternate in sign.

Proof

When $\mathcal{D} > 0$, f is strictly increasing and hence f^{-1} exists on $[f(0), f(1)]$ and is also increasing. In fact for $\mathcal{D} \neq 0$, $f^{-1}(p) = \dfrac{-vp + t}{up - s}$ for suitable p. Set $f^{-1}(p) = 0$ for $p < f(0)$ and $f^{-1}(p) = 1$ for $p > f(1)$. From Lemma 4.1 when $p < q$, $f(p) > p$ and when $p > q$, $f(p) < p$. By evaluating f^{-1} on these inequalities we obtain the required results.

When $\mathcal{D} < 0$, f is strictly decreasing and so f^{-1} exists and is decreasing. To show the convergence of the iterates we must first show that $|f'(p)| \leq \lambda < 1$ near q. The cases $u > 0$ and $u < 0$ are distinct but similar so only $u > 0$ is considered. Recall that $f'(p) = \dfrac{\mathcal{D}}{(up + v)^2}$ which is negative and increasing as $f(p)$ is convex and further $|f'(0)| = \dfrac{-\mathcal{D}}{v^2} \leq 1$ when $-\mathcal{D} \leq v^2$. When $-\mathcal{D} > v^2$ let p' denote the p where $f'(p) = -1$.

Clearly $p' = \dfrac{(\sqrt{-\mathcal{D}} - v)}{u}$ and hence

$$\begin{aligned} p' < q \quad &\Longleftrightarrow \quad \frac{(\sqrt{-\mathcal{D}} - v)}{u} < \frac{s-v}{2u} + \frac{1}{2u}\sqrt{(v-s)^2 + 4tu\}} \\ &\Longleftrightarrow \quad 2\sqrt{-\mathcal{D}} - 2v < s - v + 2\sqrt{-\mathcal{D} + \frac{1}{4}(s+v)^2} \\ &\qquad \text{(using } \sqrt{(v-s)^2 + 4tu} = \sqrt{(v+s)^2 - 4\mathcal{D}}\text{)} \end{aligned}$$

and this second inequality follows from $s + v > s + t > 0$. Hence $p' < q$ and so $\lambda = |f'(q)| < 1$. Now we use the intermediate value property.

For $p \leq p'$, $f(p) > q$. For $p > q$,

$$f(p) - q = f(p) - f(q) = f'(x)(p - q)$$

for some $x \in (q, p)$ by the intermediate value theorem. It follows that $f(p) - q$ is negative and $|f(p) - q| < \lambda(p - q)$. A similar argument establishes that $f(p) - q < |p - q|$ for $p \in (p', q)$ and hence f contracts the interval $[p', 1]$around q. The argument when $u < 0$ is similar (this time $p' > q$ and f contracts the interval $[0, p']$ around q).

Finally we must show $f(p) > (<) f^{-1}(p)$ when $p < (>) q$. To show this we apply Lemma 4.1 to the function $f \circ f$. Clearly $f \circ f$ is a fractional linear mapping (by example 4.2), it is increasing and from the properties of f we have $f \circ f(0) > f(1) > 0$, $f \circ f(q) = q$ and $f \circ f(1) < f(0) < 1$. Hence $f \circ f(p) > (<) p$ as $p < (>) q$. Now evaluate f^{-1} on this inequality to see that $f(p) < (>) f^{-1}(p)$ as $p < (>) q$.

By thinking of the graph $(p, f^{-1}(p))$ as the reflection of the graph $(p, f(p)$ in the line $y = p$ on the (p, y) plane, these inequalities provide another demonstration that $|f'(q)| < 1$.

∎

The preceding two lemmas apply directly to $L_1(p)$ and $L_2(p)$ and any compositions of them. We can use them to show results about the optimal cost function V in our search problem.

4.4 Piecewise Linearity of V

In this section we consider a policy 'search site 1 if and only if $p \geq P^*$' which we shall call a threshold policy with parameter P^*. We then calculate, in each of several cases, the corresponding value function $\overline{V}$ and show that this function satisfies the optimality equation (4.4). Hence, by the results of Chapter 2, this threshold policy is optimal. It turns out that for any specified α_i's and C_i's a large range of the possible transition matrix parameters fall into one of the cases treated in this section. This suffices to show that Ross's conjecture holds for these parameter values. We are not able to show that all parameter values fall into one of the cases and so the complete resolution of Ross's conjecture remains an open problem although we offer evidence to support its validity for all values of parameters.

The construction is split up into a small number of different classes. We shall first give some more general definitions and remarks, and then examine each case in particular, giving explicit solutions and examples. In the process of these calculations it turns out that the value function $\overline{V}$ is piecewise linear in each case.

Definitions

Let $\Delta = a - (1-b)$ and let P_1 and P_2 be values of $p \in [0,1]$ such that

$$L_1(P_1) = P_1\,, \qquad L_2(P_2) = P_2$$

respectively. Define $L_i^{k+1} = L_i \circ L_i^k$ and when $\Delta \neq 0$, $U_i = L_i^{-1}$ and $U_i^{k+1} = U_i \circ U_i^k$ for $k = 1, 2, \ldots$. Let $L_{ij} = L_j \circ L_i$, $L_\sigma = L_{\sigma_k} \circ \ldots \circ L_{\sigma_1}$ for any finite string of actions $\sigma = (\sigma_1, \ldots, \sigma_k)$ and $U_\sigma = (L_\sigma)^{-1}$, where $U_\sigma(p) = 0$ for $p < a$ and $U_\sigma(p) = 1$ for $p > 1-b$ when $\Delta > 0$ while $U_\sigma(p) = 0$ for $p > 1-b$ and $U_\sigma(p) = 1$ for $p < a$ when $\Delta < 0$.

Throughout the rest of this thesis we will also use angle brackets to denote a sequence of actions repeated a number of times. For any finite string of actions σ, let $\langle\sigma\rangle_n$ denote the string repeated n times, and let $\langle\sigma\rangle$ denote the string repeated indefinitely. Hence, $\langle 1\rangle$ denotes the strategy where at every point we search site 1, and $\langle(1,2)\rangle$ denotes the strategy $(1,2,1,2,\ldots)$.

∎

Unfortunately the natural way to describe a sequence of actions and the standard order of composition of functions run in opposite directions which is the reason for the strange definition of L_σ. Lemmas 4.1 and 4.2 show the existence and uniqueness of the P_i and the existence of the U_σ over suitable ranges of p when $\Delta \neq 0$. The arguments required to identify the corners of $\overline{V}$ are quite delicate and depend heavily on these properties of the L_i and U_i. In particular Lemma 4.1 applies directly to both L_1 and L_2 and specialises as follows.

Corollary 4.1 *(to Lemma 4.1)* The equations $L_i(p) = p$ have unique solutions $P_i \in (0,1)$,

$$P_1 = \frac{2-(\alpha_1 a + b) - \sqrt{(\alpha_1 a + b)^2 - 4\alpha_1\Delta}}{2(1-\alpha_1)},$$
$$P_2 = \frac{a - \alpha_2(2-b) + \sqrt{(a+\alpha_2 b)^2 - 4\alpha_2\Delta}}{2(1-\alpha_2)}$$

$L_i(p) > (<)\, p$ when $p < (>)\, P_i$ and further

(i) if $\Delta = 0$ then $L_1(p) = L_2(p) = a$ for all $p \in [0,1]$;

(ii) if $\Delta > 0$ then L_1 is increasing and convex, L_2 is increasing and concave and $L_2(p) > L_1(p)$ for every $p \in (0,1)$;

(iii) if $\Delta < 0$ then L_1 is decreasing and concave, L_2 is decreasing and convex and $L_2(p) < L_1(p)$ for every $p \in (0,1)$.

Proof

Lemma 4.1 applies directly to both L_1 and L_2 and by substituting parameters we find that $\mathcal{D} \equiv sv - tu = \alpha_i \Delta$ for each L_i. Further $u = \alpha_1 - 1 < 0$ for L_1 while $u = 1 - \alpha_2 > 0$ for L_2. As $L_1(0) = L_2(0) = 1 - b$ and $L_1(1) = L_2(1) = a$ the other inequalities follow directly.

∎

It is important to notice that the problem splits naturally into two classes of transition matrices, namely those with $\Delta > 0$ and those with $\Delta < 0$. This has the effect of nearly doubling the number of different cases that need to be analysed but this seems to be unavoidable.

Theorem 4.1

For the threshold policy with parameter $P^* < P_1$, the search sequence $\langle 1 \rangle$ is used on $[P^*, 1]$ when $\Delta > 0$ and on $[P^*, U_1(P^*)]$ when $\Delta < 0$.

Proof

See Figures 4.3 and 4.4 below. When $\Delta > 0$ and $p > P^*$ we see from Corollary 4.1 that $L_1(p) > p > P^*$ for $p \in (P^*, P_1)$ while from part (ii) of the Corollary $L_1(p) \geq L_1(P_1) = P_1 > P^*$ for $p \geq P_1$ so applying the threshold policy and an induction argument we see the strategy $\langle 1 \rangle$ is used for $p \in [P^*, 1]$. When $\Delta < 0$ we see from Corollary 4.1 and from part (ii) of Lemma 4.2 that L_1 and U_1 are both decreasing and $U_1(P^*) > L_1(P^*) > P_1 > P^*$. Hence $L_1(p) \geq P^*$ for $p \in [P^*, U_1(P^*)]$ (L_1 maps this interval to $[P^*, L_1(P^*)]$, while reversing the orientation of the interval, so that if $s > t$ for s, $t \in [P^*, U_1(P^*)]$ then $L_1(t) > L_1(s)$). That strategy $\langle 1 \rangle$ is used on $[P^*, U_1(P^*)]$ follows as before.

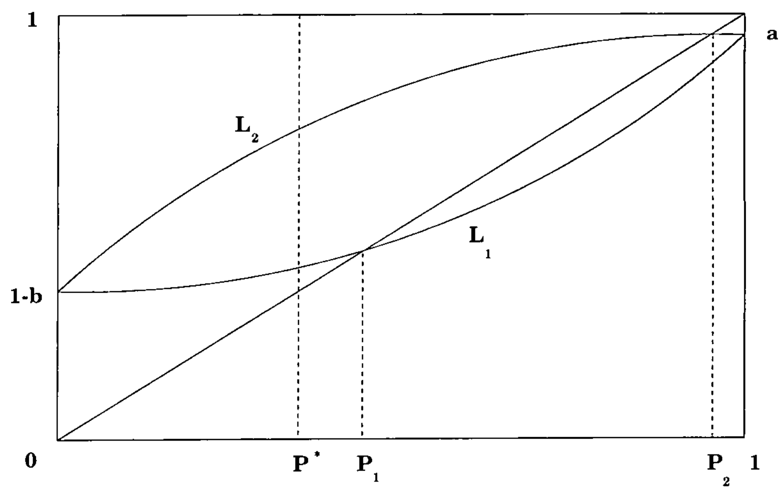


Figure 4.3: $L_1(p)$ and $L_2(p)$ in Theorem 4.1 when $\Delta > 0$

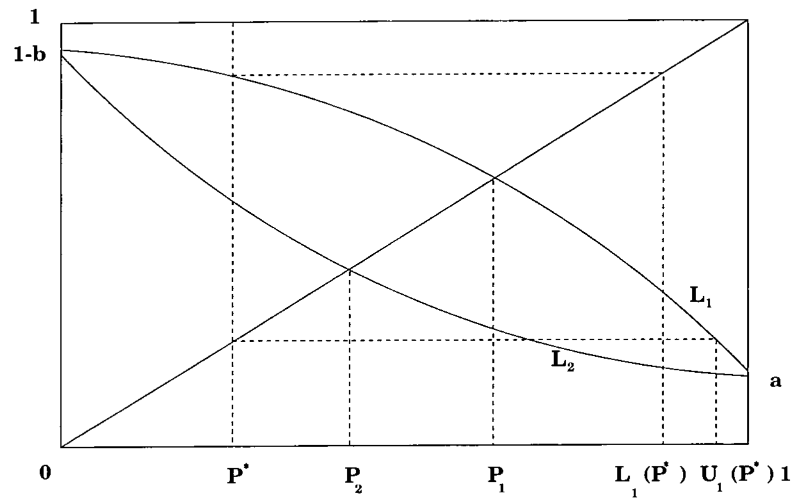


Figure 4.4: $L_1(p)$ and $L_2(p)$ in Theorem 4.1 when $\Delta < 0$

∎

The essential point here is that there is an interval of probabilities upon which L_1 is a contraction, as can be seen in figure 4.5 below. The general method applied in the next section is to find a sequence of actions σ and an interval upon which L_σ is a contraction.

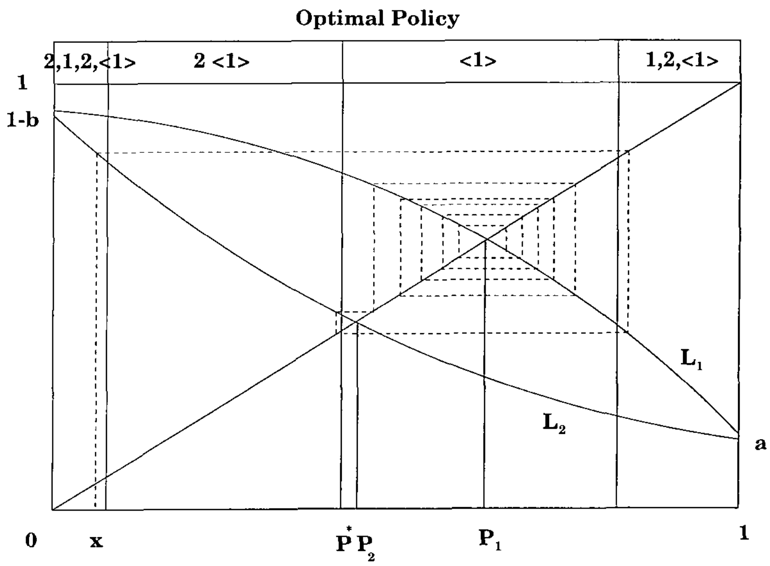


Figure 4.5: $L_1(p)$ and $L_2(p)$ in case 1 when $\Delta < 0$, showing 'cobweb' form of contraction.

<u>Aside</u> : A note on use of cobweb diagrams.

In this figure, we demonstrate how the threshold policy is determined from any point, in this case from point x. We know x is less than P^* so we have, by the definition of a threshold policy to look first in site 2. Hence, we calculate $L_2(x)$, and find that it is greater than P^*. So our next action is to search site 1, and we calculate $L_1(L_2(x))$. Continuing to use the threshold in this fashion, we can build up the strategy defined by the threshold policy for every p.

∎

Theorem 4.2

For a threshold policy with parameter $P^* < P_2$ the search sequence $\langle 2 \rangle$ is used on $[0, P^*]$ when $\Delta > 0$ and on $[U_2(P^*), P^*]$ when $\Delta < 0$.

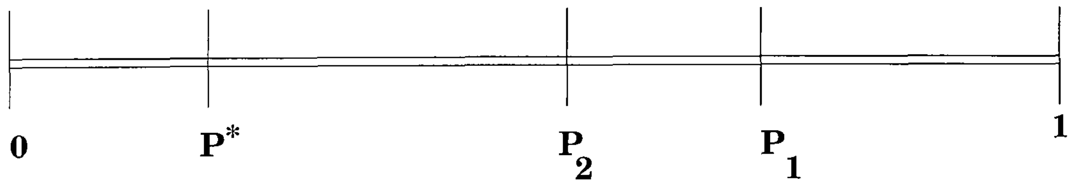
Proof

The proof is similar to that of Theorem 4.1.

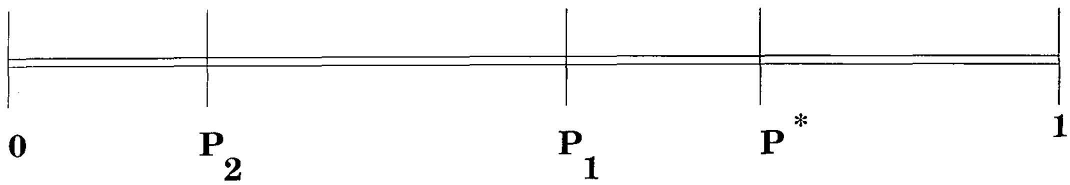
∎

We are now in a position to discuss the different classes in detail. Let us initially consider cases when $\Delta < 0$. Corollary 4.1 tells us that the L_i have unique fixed points and that $L_1(p) > L_2(p)$ for all $p \in (0,1)$. Hence, $P_1 > P_2$ for if $P_1 < P_2$ then $L_2(P_2) < L_2(P_1)$ (as L_2 is decreasing) and thus $P_2 < L_2(P_1) < L_1(P_1) = P_1$ creating a contradiction. This leaves only three distinct cases:

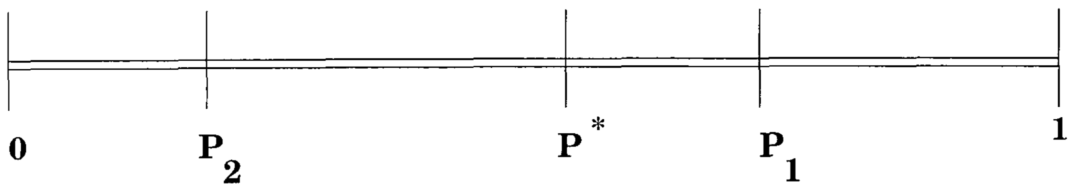
- $P_1 > P^*$ and $P_2 > P^*$



- $P_1 < P^*$ and $P_2 < P^*$

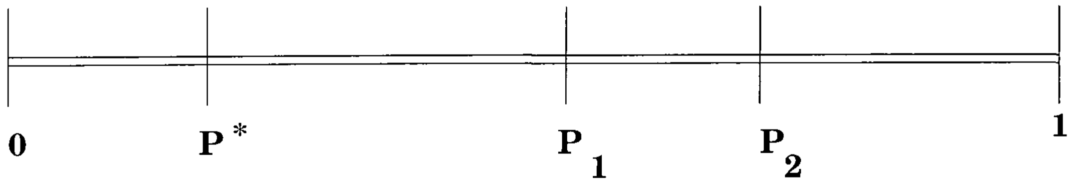


- $P_1 > P^*$ and $P_2 < P^*$

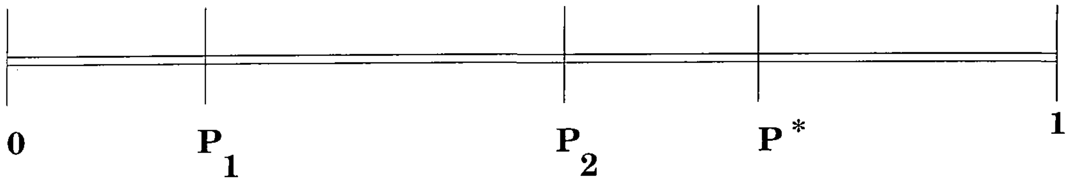


In each of these cases we demonstrate that the cost function of the threshold policy is piecewise linear and that the value of the threshold parameter P^* required to make it optimal can be explicitly calculated. When $\Delta > 0$ then $L_1(p) < L_2(p)$ and hence $P_1 < P_2$ by arguing as above. This, again results in three possible cases:

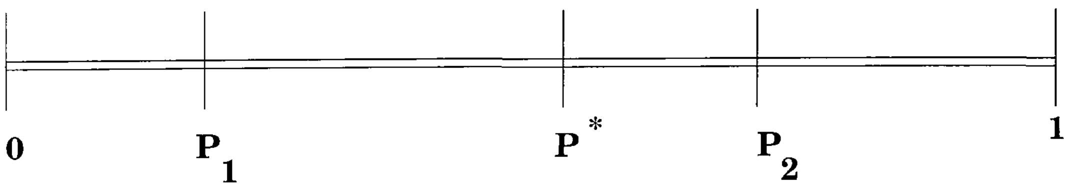
- $P_2 > P^*$ and $P_1 > P^*$



- $P_2 < P^*$ and $P_1 < P^*$



- $P_2 > P^*$ and $P_1 < P^*$



It will be seen that the first two of these cases can be dealt with when discussing the similar cases with $\Delta < 0$ so we are left with four classes into which the problem can be cleanly divided. We will demonstrate that for a large class of the parameter values the piecewise linear function $\overline{V}$, calculated using a threshold strategy with suitable parameter P^*, satisfies the optimality equation. Computer-based calculations discussed in section 4.5 strongly support the assertion that this is true for all possible parameter values.

The following notation is useful in describing the various cases. Let $Q_i^k = U_i^k(P^*)$ and similarly $Q_\sigma^k = U_\sigma^k(P^*)$ for $k = 0, 1, \ldots$ where $Q_i^0 = Q_\sigma^0 = P^*$.

4.4.1 Class 1: $P_1 > P^*$ and $P_2 > P^*$

Theorem 4.1 tells us that the threshold policy uses the strategy $\langle 1 \rangle$ at least on the interval $[P^*, U_1(P^*)]$. We proceed in this case by examining the neighbourhood immediately below P^*, solving for P^* and then finding the threshold policy for all p. Note that we cover both $\Delta > 0$ and $\Delta < 0$ in this section.

Consider $p < P^*$, and refer back to Fig. 4.3. Using the threshold strategy we look first in site 2. For $\Delta > 0$ we can apply Lemma 4.2(i) to find that $L_2(P^*) > P^*$ and further $L_2(p) \geq P^*$ for $p \in [U_2(P^*), P^*]$ so by Theorem 4.1 the threshold policy uses strategy $d = (2, \langle 1 \rangle)$ on this interval.

When $\Delta < 0$, Lemma 4.2(ii) and the inequality $L_1(p) > L_2(p)$ imply $L_2(p) > P_2 > P^*$ for all $p < P^*$ and $L_{21}(P^*) > L_2^2(P^*) > P^*$ (the latter inequality coming from $L_2(P^* < U_2(P^*) \Rightarrow P^* < L_2^2(P^*)$ as L_2 is decreasing) so that, since U_1 is decreasing, $L_2(P^*) < U_1(P^*)$ (remember the definition of L_σ). Thus the threshold policy uses strategy $(2, \langle 1 \rangle)$ for $p \in [U_{21}(P^*), P^*]$ (this interval is mapped into $[P^*, U_1(P^*)]$ by L_2 - see fig. 4.4). For any $\Delta \neq 0$ we know the strategies on an interval around P^* and can now calculate it explicitly.

Setting $V(p; \langle 1 \rangle) = A_1 p + B_1$ and $V(p; (2, \langle 1 \rangle)) = A_2 p + B_2$ we obtain the following equations:

$$\begin{aligned} A_2 p + B_2 &= C_2 + (A_1 L_2(p) + B_1)(p + \alpha_2(1-p)) \\ A_1 p + B_1 &= C_1 + (A_1 L_1(p) + B_1)(\alpha_1 p + 1 - p) \end{aligned} \tag{4.5}$$

and P^* is simply the p where these lines intersect. Solving the system for the A_i and B_i we obtain

$$A_1 = -\frac{C_1}{1-b} \qquad B_1 = \frac{C_1}{1-b}\,\frac{2-\alpha_1 a - b}{1-\alpha_1}$$

$$A_2 = \frac{C_1}{1-b}\,\frac{1-\alpha_2-\Delta(1-\alpha_1\alpha_2)}{1-\alpha_1} \qquad B_2 = C_2 + \frac{C_1}{1-b}\,\frac{\alpha_2(1-\alpha_1\Delta)}{1-\alpha_1}$$

from which we find

$$P^* = \frac{B_2 - B_1}{A_1 - A_2} = \frac{2 - \alpha_1 a - b - \alpha_2(1 - \alpha_1\Delta) - \frac{C_2}{C_1}(1-b)(1-\alpha_1)}{2 - \alpha_1 - \alpha_2 - \Delta(1 - \alpha_1\alpha_2)}$$

which in the case where $C_1 = C_2 = C$ and $\alpha_1 = \alpha_2 = \alpha$ simplifies to give

$$P^* = \frac{1 - \alpha\Delta}{2 - (1+\alpha)\Delta} \qquad (P1)$$

We can now determine all of the corners of $\overline{V}$ by working outwards from P^* in a systematic fashion.

When $\Delta > 0$ the corners are at $Q_2^{\ell+1} < \ldots < Q_2^1 < P^*$ where ℓ is such that $Q_2^{\ell+1} \leq 1 - b < Q_2^{\ell}$ (where $\ell \geq 0$ depends upon a, b and α but is finite for any choice of the parameters). There are no corners below $Q_2^{\ell+1}$ because $L_2(0) = 1 - b$ so L_2 maps $[0, Q_2^{\ell+1}]$ into $[Q_2^{\ell+1}, Q_2^{\ell}]$. We have already considered the interval $[Q_2^1, P^*]$ and any interval $[Q_2^{k+1}, Q_2^k]$ is mapped to $[Q_2^k, Q_2^{k-1}]$ by L_2. The coefficients of the piecewise linear pieces of $\overline{V}$ can be found iteratively by working outwards from the intervals adjoining P^*. Suppose that $\overline{V}(\,\cdot\,) = V(\,\cdot\,; d)$ has coefficients A_k and B_k on the interval $[Q_2^{k-1}, Q_2^{k-2}]$. They can be determined from the equation

$$A_k p + B_k = C_{d_1} + (A_{k-1} L_{d_1}(p) + B_{k-1}) P(\mathcal{U}_{d_1}) \quad .$$

(c.f. equation (4.5)). We are in the situation illustrated below:

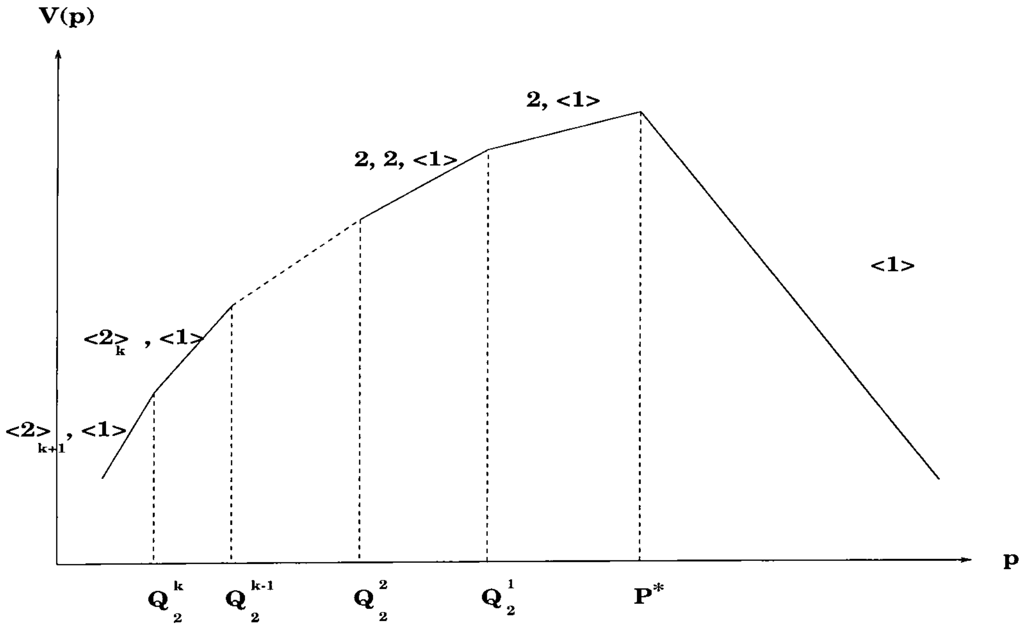


Figure 4.6: Illustration of optimal cost function in Case 1 when $\Delta > 0$

Consistency of these equations at the corners with previously calculated coefficients follows by induction using consistency at corners already treated and the fact that except at P^* strategies on adjacent intervals start with the same action. In this case, on the interval $[Q_2^{k+1}, Q_2^k]$

$$\begin{aligned} V(p\,;\,(\langle 2\rangle_{k+1}\,,\langle 1\rangle)) &= A_{k+2}p + B_{k+2} \\ &= C_2 + (p + \alpha_2(1-p))(A_{k+1}L_2(p) + B_{k+1}) \end{aligned}$$

For our induction, let us assume that we have consistency at Q_2^{k-1}, i.e., that $A_{k+1}Q_2^{k-1} + B_{k+1} = A_kQ_2^{k-1} + B_k$. Then at $p = Q_2^k$,

$$\begin{aligned} V(Q_2^k\,;\,(\langle 2\rangle_{k+1}\,,\langle 1\rangle)) &= C_2 + (Q_2^k + \alpha_2(1-Q_2^k))(A_{k+1}Q_2^{k-1} + B_{k+1}) \\ &= C_2 + (Q_2^k + \alpha_2(1-Q_2^k))(A_kQ_2^{k-1} + B_k) \qquad \text{(induction hypothesis)} \\ &= V(Q_2^k\,;\,(\langle 2\rangle_k\,,\langle 1\rangle)) \end{aligned}$$

and indeed the two lines do intersect at Q_2^k.

It remains to check that the function $\overline{V}$ constructed above satisfies the minimisation step in the optimality equation (4.4). We do this with another induction argument. Observe first that $A_1 = -C_1/(1-b) < 0$ while $A_2 > A_1$ so that $A_1p+B_1 \leq (\geq) A_2p+B_2$ precisely as $p \geq (\leq) P^*$. For $p \geq P^*$ this shows that $\overline{V}(p) = V(p\,;\,\langle 1\rangle) < V(p\,;\,(2,\langle 1\rangle))$ which, as $L_2(p) > P^*$ when $p \geq P^*$, shows that $\overline{V}$ satisfies (4.4) for these p. Similarly on $[U_1(P^*), P^*]$, $\overline{V}(p) = V(p\,;\,(2,\langle 1\rangle)) < V(p\,;\,\langle 1\rangle)$ so $\overline{V}$ satisfies (4.4) for $p \geq U_1(P^*) = Q_1^1$. Before setting up the induction hypotheses note also that

$$\begin{aligned} V(Q_1^1\,;\,(1,2,\langle 1\rangle)) &= C_1 + (\alpha_1 Q_1^1 + 1 - Q_1^1)V(P^*\,;\,(2,\langle 1\rangle)) \\ &= C_1 + (\alpha_1 Q_1^1 + 1 - Q_1^1)V(P^*\,;\,\langle 1\rangle) \\ &= V(Q_1^1\,;\,\langle 1\rangle) > V(Q_1^1\,;\,(2,\langle 1\rangle)) \quad . \end{aligned} \tag{4.6}$$

To complete the induction argument take as hypotheses

(i) $A_k > A_1$, $k \geq 2$

(ii) $V(p\,;\,(1,\langle 2\rangle_k,\langle 1\rangle)) > V(p\,;\,(\langle 2\rangle_k,\langle 1\rangle))$ for $p \in [Q_2^k, U_1(Q_2^{k-1})]$,

(iii) $V(p\,;\,(1,\langle 2\rangle_k,\langle 1\rangle)) > V(p\,;\,(\langle 2\rangle_{k+1},\langle 1\rangle))$ for $p \in [U_1(Q_2^k), Q_2^k]$.

Note that (i) is true for $k=2$, trivially, (ii) is not true for $k=0$ but equation (4.6) shows what happens in this case while (iii) does hold for $k=0$.

To establish (i) for $k > 2$ assume it is true for some $k \geq 2$, so by applying equation (3.8) we find

$$V(1;(\langle 2\rangle_{k-1},\langle 1\rangle)) - V(0;(\langle 2\rangle_{k-1},\langle 1\rangle)) = A_k > A_1$$

Hence

$$\begin{aligned} V(1\,;\,(\langle 2\rangle_k,\langle 1\rangle)) &- V(0\,;\,(\langle 2\rangle_k,\langle 1\rangle)) = A_{k+1} \\ &= V(a\,;\,(\langle 2\rangle_{k-1},\langle 1\rangle)) - \alpha_2 V(1-b\,;\,(\langle 2\rangle_{k-1},\langle 1\rangle)) \end{aligned}$$

$$\begin{aligned} &> V(a\,;\,(\langle 2\rangle_{k-1}\,,\langle 1\rangle)) - V(1-b\,;\,(\langle 2\rangle_{k-1}\,,\langle 1\rangle)) \\ &= \Delta[V(1\,;\,(\langle 2\rangle_{k-1}\,,\langle 1\rangle)) - V(0\,;\,(\langle 2\rangle_{k-1}\,,\langle 1\rangle))] \\ &= \Delta A_k > A_1 \end{aligned}$$

as if $A_k > 0$,then $\Delta A_k > 0 > A_1$, while if $A_k < 0$, then $\Delta A_k > A_k > A_1$ and so result (i) holds for all k. For part (ii) with $k \geq 1$ we proceed as in (4.6) to get

$$V(U_1(Q_2^k)\,;\,(1,\langle 2\rangle_{k+1}\,,\langle 1\rangle)) = V(U_1(Q_2^k)\,;\,(1,\langle 2\rangle_{k}\,,\langle 1\rangle)) > V(U_1(Q_2^k)\,;\,(\langle 2\rangle_{k+1}\,,\langle 1\rangle))$$

by hypothesis (iii). In addition,

$$\begin{aligned} &V(0\,;\,(1,\langle 2\rangle_{k+1}\,,\langle 1\rangle)) - V(0\,;\,(\langle 2\rangle_{k+1}\,,\langle 1\rangle)) \\ &\quad = C_1 + (1-b)[V(1\,;\,(\langle 2\rangle_{k+1}\,,\langle 1\rangle)) - V(0\,;\,(\langle 2\rangle_{k+1}\,,\langle 1\rangle))] \\ &\quad > C_1 + (1-b)A_1 = 0 \end{aligned}$$

by hypothesis (i) and these are sufficient to establish (ii) for $k+1$ because $V(p\,;\,\boldsymbol{d})$ is linear in p for any $\boldsymbol{d}$. Finally, as (3.6) implies that $V(p\,;\,(1,\boldsymbol{d})) - V(p\,;\,(2,\boldsymbol{d}))$ is decreasing in p for any $\boldsymbol{d}$ and

$$V(Q_2^{k+1}\,;\,(\langle 2\rangle_{k+2}\,,\langle 1\rangle)) = V(Q_2^{k+1}\,;\,(\langle 2\rangle_{k+1}\,,\langle 1\rangle)) < V(Q_2^{k+1}\,;\,(1,\langle 2\rangle_{k+1}\,,\langle 1\rangle))$$

by consistency at the corners and part (ii), we see that (iii) holds for $p \leq Q_2^{k+1}$. The constructed function $\overline{V}$ thus satisfies the optimality equation (4.4) and so equals the minimal expected detection cost. It follows that we have shown that the optimal strategy on $[Q_2^k\,, Q_2^{k-1}]$ is $(\langle 2\rangle_k, \langle 1\rangle)$, stopping when the target is first seen of course.

When $\Delta = 0$ the preceding analysis holds with the interpretation that $Q_1^1 = Q_2^1 = 0$ so that $\overline{V}$ has only a single corner at $P^* \leq P_1 = P_2 = a$.

When $\Delta < 0$ things are a little different as unsuccessful searches do not have the same effect on our belief about the target's whereabouts. Recall that the threshold strategy is $\langle 1\rangle$ for $p \in [P^*, U_1(P^*)]$ and $(2, \langle 1\rangle)$ on $[Q_{21}^1\,, P^*]$, which is mapped into $[P^*, U_1(P^*)]$ by L_2. In this case we find successive intervals alternately above and below P^*.

The corners are at the points

$$Q_{21}^{\ell+1} < \ldots < Q_{21}^{1} < P^{*} < U_1(P^{*}) < U_1(Q_{21}^{1}) < \ldots < U_1(Q_{21}^{\ell})$$

(where $Q_{21}^{\ell+1} \leq a < Q_{21}^{\ell}$. Note that L_1 maps $[U_1(Q_{21}^{\ell}), 1]$ into $[Q_{21}^{\ell+1}, Q_{21}^{\ell}]$). L_2 maps an interval $[Q_{21}^{k}, Q_{21}^{k-1}]$ over to $[U_1(Q_{21}^{k-2}), U_1(Q_{21}^{k-1})]$ which is mapped in turn by L_1 over to $[Q_{21}^{k-1}, Q_{21}^{k-2}]$. Consistency of the coefficients at the corners can be checked as before. The optimal strategy on $[Q_{21}^{k}, Q_{21}^{k-1}]$ is $(\langle 2,1\rangle_k, \langle 1\rangle)$. On $[U_1(Q_{21}^{k-2}), U_1(Q_{21}^{k-1})]$ the optimal strategy is $(\langle 1,2\rangle_k, \langle 1\rangle)$ (the required actions can be read off the list of subscripts on the symbols describing the corners). First we need to assign parameters to the individual arcs of $\overline{V}$, and we do this as in figure 4.7 below.

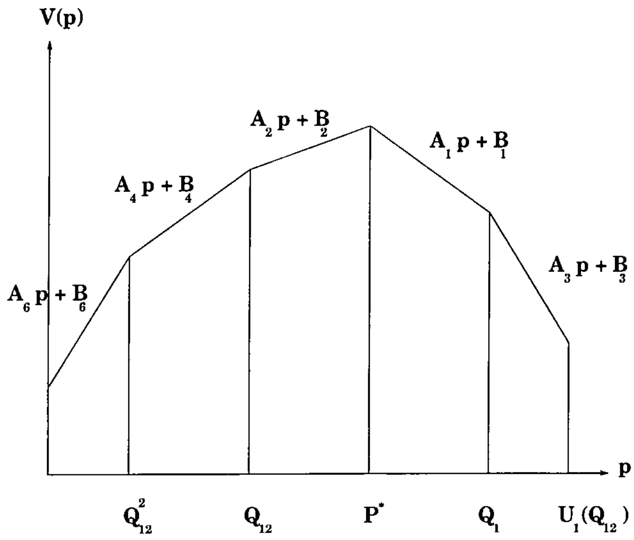


Figure 4.7: Form of the optimal cost function $\overline{V}(p)$ when P_1, $P_2 \geq P^{*}$ and $\Delta < 0$

We know that we have consistency at P^*, so let us begin by considering the arc $A_3p + B_3$. For p in the segment $[Q_1, U_1(Q_{21})]$

$$\begin{aligned}\overline{V}(p) = V(p; 1, 2, \langle 1\rangle) &= C_1 + (\alpha_1 p + (1-p))\overline{V}(L_1(p)) \\ &= C_1 + (\alpha_1 p + (1-p))(A_2 L_1(p) + B_2)\end{aligned}$$

And at $U_1(P^*)$ we find

$$\begin{aligned}V(U_1(P^*); 1, 2, \langle 1\rangle) &= C_1 + (\alpha_1 U_1(P^*) + (1 - U_1(P^*)))(A_2 P^* + B_2) \\ &= C_1 + (\alpha_1 U_1(P^*) + (1 - U_1(P^*)))(A_1 P^* + B_1) \\ &= V(U_1(P^*); \langle 1\rangle)\end{aligned}$$

so we have consistency at $U_1(P^*)$. Now we look at $A_4p + B_4$ on the segment $[Q_{21}^2, Q_{21}]$

$$\begin{aligned}\overline{V}(p) = V(p; 2, 1, 2, \langle 1\rangle) &= C_2 + (\alpha_2(1-p) + p)\overline{V}(L_2(p)) \\ &= C_2 + (\alpha_2(1-p) + p)(A_3 L_2(p) + B_3)\end{aligned}$$

and at $p = Q_{21}$ we find

$$\begin{aligned}V(Q_{21}; 2, 1, 2, \langle 1\rangle) &= C_2 + (\alpha_2(1 - Q_{21}) + Q_{21})(A_3 U_1(P^*) + B_3) \\ &= C_2 + (\alpha_2(1 - Q_{21}) + Q_{21})(A_1 U_1(P^*) + B_1) \\ &= V(Q_{21}; 2, \langle 1\rangle)\end{aligned}$$

and we have consistency at Q_{21}. We can complete the consistency argument by induction. Let us assume that we have consistency at some Q_{21}^k i.e., the arcs $A_{2k}p+B_{2k}$ and $A_{2k+2}p+B_{2k+2}$ intersect at $p = Q_{21}^k$. Now we consider arc $A_{2k+3}p+B_{2k+3}$ on $[U_1(Q_{21}^k), U_1(Q_{21}^{k+1})]$.

$$\begin{aligned}V(p; \langle 1, 2\rangle_{k+1}, \langle 1\rangle) &= A_{2k+3}p + B_{2k+3} \\ &= C_1 + (\alpha_1 p + (1-p))\overline{V}(L_1(p)) \\ &= C_1 + (\alpha_1 p + (1-p))(A_{2k+2} L_1(p) + B_{2k+2})\end{aligned}$$

so at $p = U_1(Q_{21}^k)$, we find

$$\begin{aligned} V(U_1(Q_{21}^k); \langle 1,2\rangle_{k+1}, \langle 1\rangle) &= C_1 + (\alpha_1 U_1(Q_{21}^k) + (1 - U_1(Q_{21}^k)))(A_{2k+2}Q_{21}^k + B_{2k+2} \\ &= C_1 + (\alpha_1 U_1(Q_{21}^k) + (1 - U_1(Q_{21}^k)))(A_{2k}Q_{21}^k + B_{2k}) \\ &= V(U_1(Q_{21}^k); \langle 1,2\rangle_k, \langle 1\rangle) \end{aligned}$$

and we have consistency at $U_1(Q_{21}^k)$. To complete the proof, we consider arc $A_{2k+4}p + B_{2k+4}$ on $[Q_{21}^{k+2}, Q_{21}^{k+1}]$.

$$\begin{aligned} V(p; 2, \langle 1,2\rangle_{k+1}, \langle 1\rangle) &= A_{2k+4}p + B_{2k+4} \\ &= C_2 + (\alpha_2(1-p) + p)\overline{V}(L_2(p)) \\ &= C_1 + (\alpha_2(1-p) + p)(A_{2k+3}L_2(p) + B_{2k+3}) \end{aligned}$$

and at $p = Q_{21}^{k+1}$ we discover

$$\begin{aligned} V(Q_{21}^{k+1}; 2, \langle 1,2\rangle_{k+1}, \langle 1\rangle) &= C_2 + (\alpha_2(1 - Q_{21}^{k+1}) + Q_{21}^{k+1})(A_{2k+3}U_1(Q_{21}^k) + B_{2k+3} \\ &= C_2 + (\alpha_2(1 - Q_{21}^{k+1}) + Q_{21}^{k+1})(A_{2k+1}U_1(Q_{21}^k) + B_{2k+1}) \\ &= V(Q_{21}^{k+1}; 2, \langle 1,2\rangle_k, \langle 1\rangle) \end{aligned}$$

and so we have consistency at Q_{21}^{k+1}. Hence by induction we have consistency at all corners. We can now show that the function calculated $\overline{V}$ satisfies (4.4). To do this we first show that $\overline{V}$ is a concave function so $\ldots A_{2k+2} < A_{2k} < \ldots < A_4 < A_2 < A_1 < A_3 < \ldots < A_{2n-1} < A_{2n+1} \ldots$ This is easiest done by induction following examination of the transition structure we have in this case. We know that arc $A_5p + B_5$ is mapped to $A_4p + B_4$ and then to $A_3p + B_3$ and so on. Consider then first the trio of arcs of $\overline{V}$ $A_3p + B_3$, $A_2p + B_2$ and $A_1p + B_1$. We can show the relationship between them as below

$$\begin{pmatrix} A_3 \\ B_3 \end{pmatrix} = \begin{pmatrix} 0 \\ C_1 \end{pmatrix} + \begin{pmatrix} \alpha_1 a + b - 1 & \alpha_1 - 1 \\ 1 - b & 1 \end{pmatrix} \begin{pmatrix} A_2 \\ B_2 \end{pmatrix}$$

$$\begin{pmatrix} A_1 \\ B_1 \end{pmatrix} = \begin{pmatrix} 0 \\ C_1 \end{pmatrix} + \begin{pmatrix} \alpha_1 a + b - 1 & \alpha_1 - 1 \\ 1 - b & 1 \end{pmatrix} \begin{pmatrix} A_1 \\ B_1 \end{pmatrix}$$

and we also know from consistency that

$$\begin{aligned} A_3 Q_1^1 + B_3 &= A_1 Q_1^1 + B_1 \\ A_2 P^* + B_2 &= A_1 P^* + B_1 \end{aligned}$$

from which we can deduce

$$\begin{pmatrix} A_1 - A_3 \\ B_1 - B_3 \end{pmatrix} = \begin{pmatrix} \alpha_1 a + b - 1 & \alpha_1 - 1 \\ 1 - b & 1 \end{pmatrix} \begin{pmatrix} A_1 - A_2 \\ B_1 - B_2 \end{pmatrix}$$

which with substitution gives

$$(A_1 - A_3) \begin{pmatrix} 1 \\ -Q_1^1 \end{pmatrix} = (A_1 - A_2) \begin{pmatrix} \alpha_1 a + b - 1 & \alpha_1 - 1 \\ 1 - b & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -P^* \end{pmatrix}$$

so

$$-Q_1^1 (A_1 - A_3) = (A_1 - A_2)(1 - b - P^*)$$

and so $A_3 < A_1$ as $A_1 < A_2$ and $P^* < P_1 \leq 1 - b$ For our inductive hypothesis, suppose $A_{2k-1} > A_{2k+1}$. Then, as above

$$\begin{aligned} \begin{pmatrix} A_{2k+2} \\ B_{2k+2} \end{pmatrix} &= \begin{pmatrix} 0 \\ C_2 \end{pmatrix} + \begin{pmatrix} a + \alpha_2 b - \alpha_2 & 1 - \alpha_2 \\ \alpha_2 (1 - b) & \alpha_2 \end{pmatrix} \begin{pmatrix} A_{2k+1} \\ B_{2k+1} \end{pmatrix} \\ \begin{pmatrix} A_{2k} \\ B_{2k} \end{pmatrix} &= \begin{pmatrix} 0 \\ C_2 \end{pmatrix} + \begin{pmatrix} a + \alpha_2 b - \alpha_2 & 1 - \alpha_2 \\ \alpha_2 (1 - b) & \alpha_2 \end{pmatrix} \begin{pmatrix} A_{2k-1} \\ B_{2k-1} \end{pmatrix} \end{aligned}$$

and from consistency

$$\begin{aligned} A_{2k+2} Q_{21}^k + B_{2k+2} &= A_{2k} Q_{21}^k + B_{2k} \\ A_{2k+1} U_1 (Q_{21}^{k-1}) + B_{2k+1} &= A_{2k-1} U_1 (Q_{21}^{k-1}) + B_{2k-1} \end{aligned}$$

So we find by substitution

$$(A_{2k+2} - A_{2k}) \begin{pmatrix} 1 \\ -Q_{21}^k \end{pmatrix} = (A_{2k+1} - A_{2k-1}) \begin{pmatrix} a + \alpha_2 b - \alpha_2 & 1 - \alpha_2 \\ \alpha_2 (1 - b) & \alpha_2 \end{pmatrix} \begin{pmatrix} 1 \\ -U_1 (Q_{21}^{k-1}) \end{pmatrix}$$

and so we find

$$-Q_{21}^k(A_{2k+2}-A_{2k}) = (\alpha_2(1-b)-\alpha_2 U_1(Q_{21}^{k-1}))(A_{2k+1}-A_{2k-1})$$

and hence $A_{2k+2} > A_{2k}$ as $A_{2k+1} < A_{2k-1}$ and $U_1(Q_{21}^{k-1}) < 1-b$ by the definition of the corners.

To complete the proof, we now take as our induction hypothesis that $A_{2k+2} > A_{2k}$. Then as before,

$$\begin{pmatrix} A_{2k+3} \\ B_{2k+3} \end{pmatrix} = \begin{pmatrix} 0 \\ C_1 \end{pmatrix} + \begin{pmatrix} \alpha_1 a + b - 1 & \alpha_1 - 1 \\ 1-b & 1 \end{pmatrix} \begin{pmatrix} A_{2k+2} \\ B_{2k+2} \end{pmatrix}$$
$$\begin{pmatrix} A_{2k+1} \\ B_{2k+1} \end{pmatrix} = \begin{pmatrix} 0 \\ C_1 \end{pmatrix} + \begin{pmatrix} \alpha_1 a + b - 1 & \alpha_1 - 1 \\ 1-b & 1 \end{pmatrix} \begin{pmatrix} A_{2k} \\ B_{2k} \end{pmatrix}$$

and, by consistency

$$\begin{aligned} A_{2k+3}U_1(Q_{21}^k) + B_{2k+3} &= A_{2k+1}U_1(Q_{21}^k) + B_{2k+1} \\ A_{2k+2}Q_{21}^k + B_{2k+2} &= A_{2k}Q_{21}^k + B_{2k} \end{aligned}$$

so by substitution, we find

$$(A_{2k+1} - A_{2k+3}) \begin{pmatrix} 1 \\ -U_1(Q_{21}^k) \end{pmatrix} = (A_{2k} - A_{2k+2}) \begin{pmatrix} \alpha_1 a + b - 1 & \alpha_1 - 1 \\ 1-b & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -Q_{21}^k \end{pmatrix}$$

and so

$$-U_1(Q_{21}^k)(A_{2k+1} - A_{2k+3}) = (A_{2k} - A_{2k+2})(1 - b - Q_{21}^k)$$

and so $A_{2k+1} > A_{2k+3}$ as $1 - b - Q_{21}^k > 1 - b - P^* > 0$. Thus the result is shown.

To show the minimisation step is satisfied we use another induction argument. For simplicity, we will only show the argument for $p < P^*$ as the result follows similarly for $p > P^*$. Let us begin by considering the three arcs $A_3p + B_3$, $A_1p + B_1$ and $A_2p + B_2$. For $p \in [Q_{21}, P^*]$ we want to show that $C_1 + (\alpha_1 p + 1 - p)V(L_1(p)) > C_2 + (p + \alpha_2(1 - p))(V(L_2(p))$. We know from Lemma 4.2 that for $p < P^*$, $U_1(p) > L_1(p) > P_1$. Hence taking first action 1 maps us to a state $p' > P^*$. In fact, the segment we are interested in, (Q_{21}, P^*), is mapped to $(L_2(P^*), U_1(P^*))$ and $(L_1(P^*), L_1(Q_{21}))$ by actions 2 and 1 respectively, where $L_1(P^*) \in (P^*, U_1(P^*))$ and $L_1(Q_{21}) \in (U_1(P^*), U_1(Q_{21}))$. We want to show that for $p \in (Q_{21}, P^*)$ that action 2 first is optimal. To do this we examine the two parts of the segment individually. On interval (Q_1^2, P^*) we find from (4.4)

$$\begin{aligned}\overline{V}(p) &= \min\{C_1 + (\alpha_1 p + (1-p))(A_1 L_1(p) + B_1); C_2 + (p + \alpha_2(1-p))(A_1 L_2(p) + B_1)\} \\ &= \min\{V(p;\langle 1\rangle), V(p;2,\langle 1\rangle)\} \\ &= V(p;2,\langle 1\rangle)\end{aligned}$$

where the last result comes from the concavity of $\overline{V}$ and consistency at P^*. So now we consider (Q_{21}, Q_1^2) which is mapped by action 2 into $(L_2(Q_1^2), Q_1)$ on which $\overline{V}(p) = A_1 p + B_1$ and by action 1 into $(Q_1, L_1(Q_{21}))$ on which $\overline{V}(p) = A_3 p + B_3$. Hence we are comparing $V(p;2,\langle 1\rangle)$ with $V(p;1,1,2,\langle 1\rangle)$. At $p = Q_1^2$, we find

$$\begin{aligned}V(Q_1^2;1,1,2,\langle 1\rangle) &= C_1 + (\alpha_1 Q_1^2 + (1-Q_1^2))(A_3 Q_1 + B_3) \\ &= C_1 + (\alpha_1 Q_1^2 + (1-Q_1^2))(A_1 Q_1 + B_1) \\ &= V(Q_1^2;\langle 1\rangle) \\ &> V(Q_1^2;2,\langle 1\rangle)\end{aligned}$$

To complete the proof on this segment, we need to show that $V(p;1,1,2,\langle 1\rangle) > V(p;2,\langle 1\rangle)$ for $p < Q_1^2$. We do this by showing that the gradient of $V(p;1,1,\boldsymbol{d})$ is less than the gradient of $V(p;\boldsymbol{d})$ for any strategy $\boldsymbol{d}$ for which $V(p;\boldsymbol{d})$ has positive gradient.

Let $V(p;\boldsymbol{d}) = Ap + B$ with $A > 0$, $B > 0$ and let $V(p;1,1,\boldsymbol{d}) = A'p + B'$. Applying the action 1 map twice we get

$$\begin{aligned}A' &= (\alpha_1 a + b - 1)^2 A + (\alpha_1 a + b - 1)(\alpha_1 - 1)B + (\alpha_1 - 1)[C_1 + (1-b)A + B] \\ &= (\alpha_1 a + b - 1)^2 A + (\alpha_1 - 1)[C_1 + (1-b)A + (\alpha_1 a + b)B] \\ &< (\alpha_1 a + b - 1)^2 A \quad < \quad A\end{aligned}$$

and the result follows. We are now in a position to complete the argument. Take as induction hypothesis that $\overline{V}$ satisfies the optimality equation on (Q_{21}^k, Q_{21}^{k-1}), so in particular $V(Q_{21}^k; 1, 1, \langle 2, 1\rangle, \langle 1\rangle) > V(Q_{21}^k; \langle 2, 1\rangle, \langle 1\rangle)$ and consider a segment (Q_{21}^{k+1}, Q_{21}^k). As above we break the segment up into two sub-sections at $U_1^2(Q_{21}^k)$. On the first, $(U_1^2(Q_{21}^k), Q_{21}^k)$, we must compare $V(p; \langle 2, 1\rangle_k, \langle 1\rangle)$ with $V(p; 1, 1, \langle 2, 1\rangle_k, \langle 1\rangle)$. By the above argument we know that $V(p; 1, 1, d) - V(p; d)$ decreases as p increases. From the induction hypothesis $V(Q_{21}^k; \langle 2, 1\rangle_k, \langle 1\rangle) > V(Q_{21}^k; 1, 1, \langle 2, 1\rangle_k, \langle 1\rangle)$ so the inequality is also true for $p < Q_{21}^k$. On the second sub-segment, we must compare $V(p; \langle 2, 1\rangle_k, \langle 1\rangle)$ with $V(p; 1, 1, \langle 2, 1\rangle_{k+1}, \langle 1\rangle) = V(p; 1, \langle 1, 2\rangle_{k+1}, \langle 1\rangle)$. By consistency at corner $U_1(Q_{21}^k)$, $V(U_1^2(Q_{21}^k); 1, \langle 1, 2\rangle_{k+1}, \langle 1\rangle) = V(U_1^2(Q_{21}^k); 1, \langle 1, 2\rangle_k, \langle 1\rangle) > V(U_1^2(Q_{21}^k); 1, \langle 2, 1\rangle_k, \langle 1\rangle)$ by the argument for the previous sub-segment. Repeating that argument,

$$V(p; 1, 1, \langle 2, 1\rangle_{k+1}, \langle 1\rangle) > V(p; \langle 2, 1\rangle_{k+1}, \langle 1\rangle)$$

for all $p < U_1^2(Q_{21}^k)$. Hence, by induction, $\overline{V}$ satisfies the minimisation step for $p < P^*$. A similar argument works for $p > P^*$.

Hence, if $P_1 > P^*$ and $P_2 > P^*$, with P^* as given above, we have found the form of $V(p)$ for all $p \in [0, 1]$.

Example 4.3

To get an example of this case take $a = 0.3$, $b = 0.01$, $\alpha_1 = 0.7$, $\alpha_2 = 0.75$, $C_1 = 1.4$ and $C_2 = 1.5$. With these values we find that $P_1 = 0.6212$, $P_2 = 0.5575$ and using the expression above $P^* = 0.3982$ and the conditions describing class 1 are satisfied. We can approximate the optimal cost function $\overline{V}$ using the Matlab program **diffalpha.m** which can be found in the appendices. Using this program, we find the following graphs. This example clearly shows the L_i are of the form described in Corollary 4.1 and that V is piecewise linear in p.

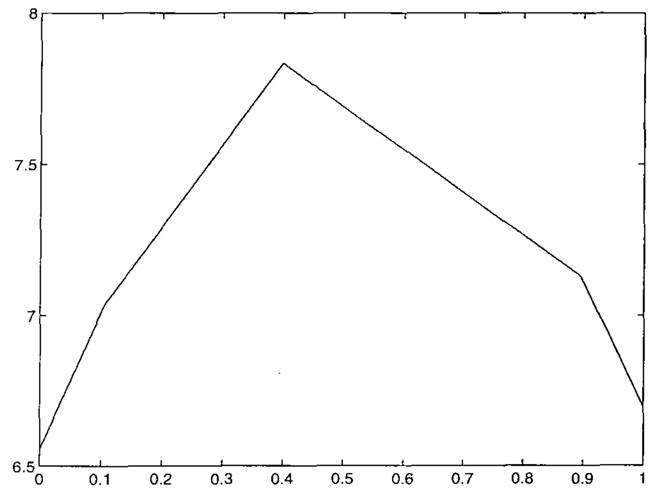


Figure 4.8: Optimal cost function $\overline{V}(p)$ for Class 1 with parameter values as in Example 4.3

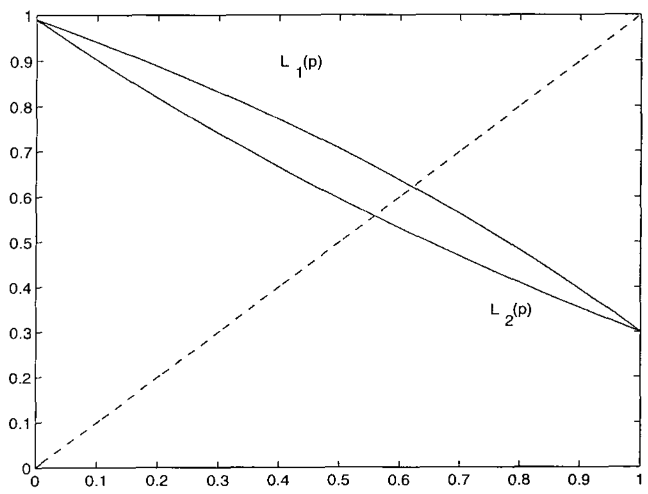


Figure 4.9: $L_1(p)$ and $L_2(p)$ with parameter values as in example 4.3

4.4.2 Class 2: $P_1 < P^*$ and $P_2 < P^*$

The same arguments that appeared in subsection 4.4.1 can be applied to this situation, with reference to Figures 4.9, 4.10 below.

When $\Delta > 0$ the threshold strategy for $p \in [0, P^*]$ is $\langle 2 \rangle$ and for $p \in [Q_1^{k-1}, Q_1^k]$ it is $(\langle 1 \rangle_k, \langle 2 \rangle)$. $\overline{V}$ has corners at $P^* < Q_1^1 < \ldots < Q_1^{r+1}$ where r is such that $Q_1^r < a \leq Q_1^{r+1}$.

When $\Delta = 0$ the threshold strategy $\langle 2 \rangle$ is used for $p \in [0, P^*]$ while $(1, \langle 2 \rangle)$ is used for $p \geq P^* \geq a$.

When $\Delta < 0$ the strategy $\langle 2 \rangle$ is optimal for $p \in [U_2(P^*), P^*]$ while $(1, \langle 2 \rangle)$ is optimal for $p \in [P^*, Q_{12}^1]$. This time $\overline{V}$ has corners at $U_2(Q_{12}^r) < \ldots < U_2(P^*) < P^* < Q_{12}^1 < \ldots < Q_{12}^{r+1}$.

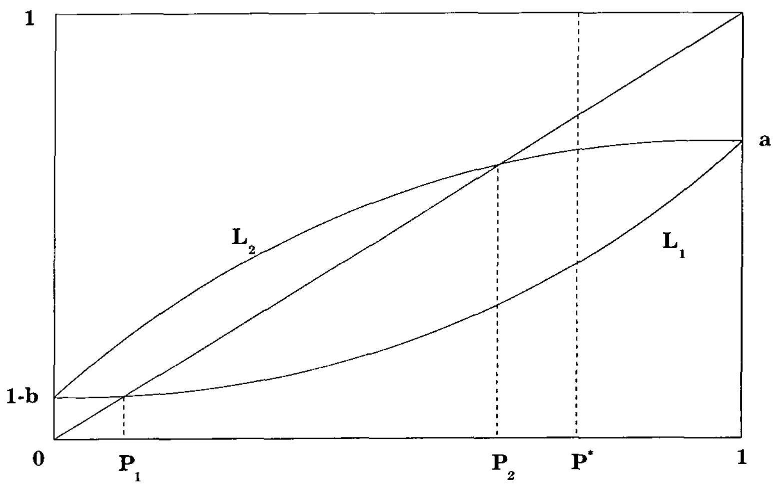


Figure 4.10: $L_1(p)$ and $L_2(p)$ in Case 2 when $\Delta > 0$

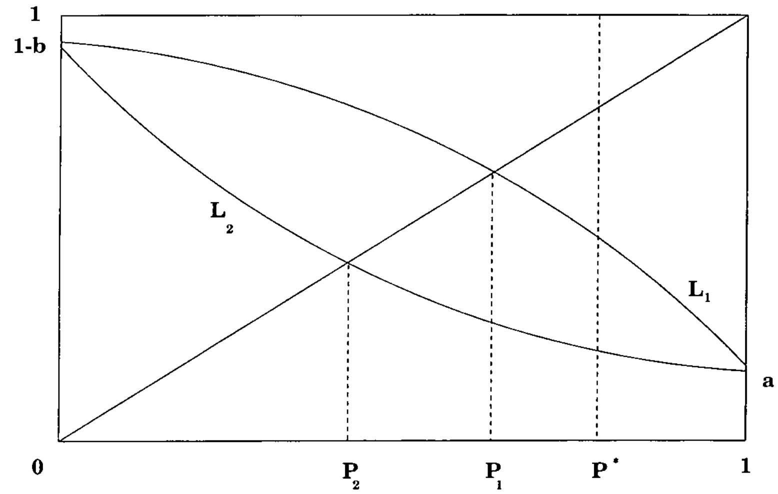


Figure 4.11: $L_1(p)$ and $L_2(p)$ in Case 2 when $\Delta < 0$

The value of P^* can again be explicitly calculated and using a similar notation to case 1 we find

$$A_1 = -\frac{C_2}{1-a}\frac{1-\alpha_1-\Delta(1-\alpha_1\alpha_2)}{1-\alpha_2} \qquad B_1 = C_1 + \frac{C_2}{1-a}\frac{1-\Delta}{1-\alpha_2}$$

$$A_2 = \frac{C_2}{1-a} \qquad B_2 = \frac{C_2}{1-a}\frac{1-a+\alpha_2(1-b)}{1-\alpha_2}$$

and hence that

$$P^* = \frac{(1-\alpha_2)(1-b)+\frac{C_1}{C_2}(1-\alpha_2)(1-a)}{2-\alpha_1-\alpha_2-\Delta(1-\alpha_1\alpha_2)}$$

which simplifies in the case where $\alpha_i = \alpha$ and $C_i = C$ to give

$$P^* = \frac{1-\Delta}{2-(1+\alpha)\Delta} \quad . \qquad (P2)$$

Example 4.4

An example of this class can be found by setting $a = 0.001$, $b = 0.1$, $\alpha_1 = 0.5$, $\alpha_2 = 0.4$, $C_1 = 1.2$ and $C_2 = 1$. From this $P_1 = 0.5548$, and $P_2 = 0.3675$. This produces a value of $P^* = 0.6922$. The optimal cost function solution in this case is shown in figure 4.12 below.

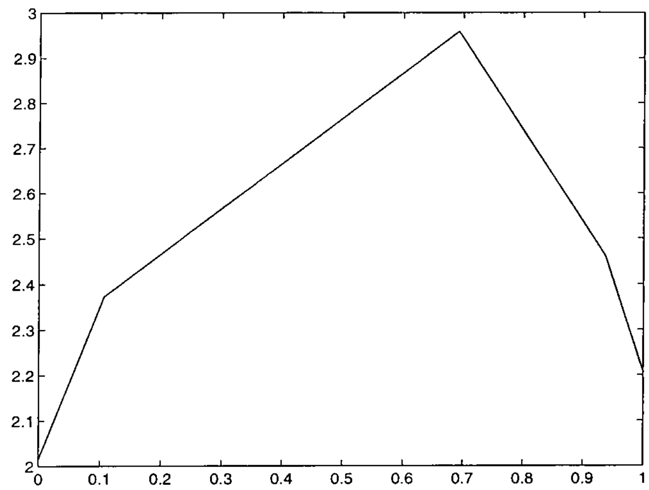


Figure 4.12: Optimal cost function $\overline{V}(p)$ for Class 2 with parameter values as in Example 4.4

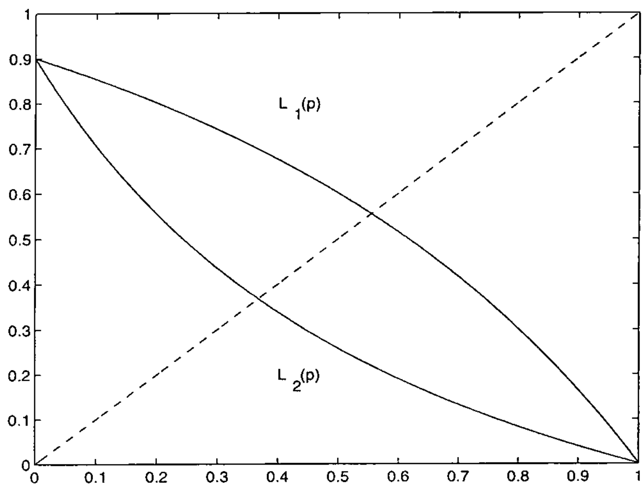


Figure 4.13: $L_1(p)$ and $L_2(p)$ with parameter values as in Example 4.4

∎

4.4.3 Class 3: $P_1 > P^*$ and $P_2 < P^*$

This is the most straightforward of the four cases. By Lemma 4.2 we see that $\Delta < 0$ and by Theorems 4.1 and 4.2 above it is clear (c.f. Fig. 4.14) that threshold strategy $\langle 1\rangle$ is used for $p \in [P^*, U_1(P^*)]$ and threshold strategy $\langle 2\rangle$ is used for $p \in [U_2(P^*), P^*]$ and arguing as in cases 1 and 2, $\overline{V}$ has corners at

$$\ldots < Q_{21}^2 < U_2(Q_{12}^1) < Q_{21}^1 < Q_2^1 < P^* < Q_1^1 < Q_{12}^1 < U_1(Q_{21}^1) < Q_{12}^2 < \ldots$$

where for example, L_2 maps $[Q_{21}^{k+1}, U_2(Q_{12}^k)]$ to $[Q_{12}^k, U_1(Q_{21}^k)]$. The threshold strategy on $[Q_{21}^{k+1}, U_2(Q_{12}^k)]$ is $(\langle 2,1\rangle_{k+1}, \langle 1\rangle)$ and the strategies on the other intervals can be read off in the usual way.

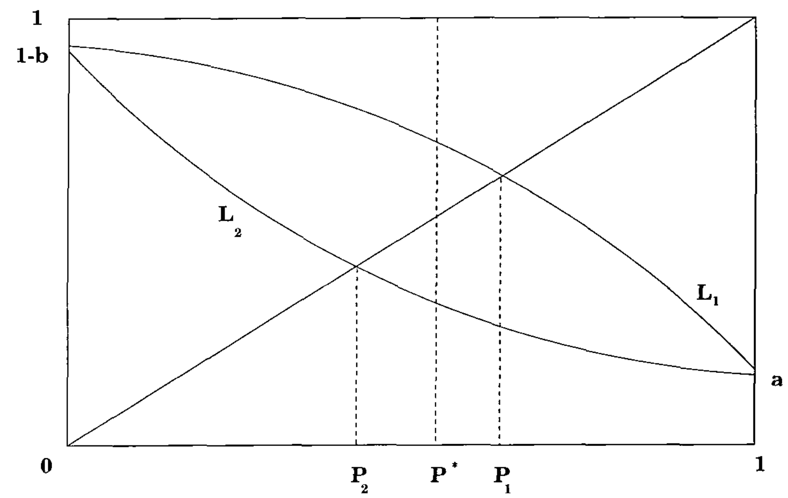


Figure 4.14: $L_1(p)$ and $L_2(p)$ in Case 3

We can calculate P^* just as in the earlier cases and we find, if A_1p+B_1 and A_2p+B_2

represent the arcs of $\overline{V}(p)$ which intersect at P^* that

$$A_1 = -\frac{C_1}{1-b} \qquad B_1 = \frac{C_1}{1-b}\frac{2-b-a\alpha_1}{1-\alpha_1}$$

$$A_2 = \frac{C_2}{1-a} \qquad B_2 = \frac{C_2}{1-a}\frac{1-a+\alpha_2(1-b)}{1-\alpha_2}$$

and hence

$$P^* = \frac{C_1(1-\alpha_2)(ab+(1-a)(1-\alpha_1 a)-\Delta)+C_2(1-\alpha_1)(\Delta-ab-\alpha_2(b-1)^2)}{(C_1(1-a)+C_2(1-b))(1-\alpha_1)(1-\alpha_2)}$$

which simplifies in the case $\alpha_i = \alpha$ and $C_i = C$ to give

$$P^* = \frac{1-a}{2-a-b} + \frac{\alpha}{1-\alpha}(b-a) \qquad (P3)$$

Example 4.5

Taking $a = 0.3$, $b = 0.001$, $\alpha_1 = 0.5$, $\alpha_2 = 0.4$, $C_1 = 1$ and $C_2 = 0.9$ we find that $P_1 = 0.6570$, and $P_2 = 0.4998$, and $P^* = 0.5884$. The graphs are as shown in figures 4.15 and 4.16 below.

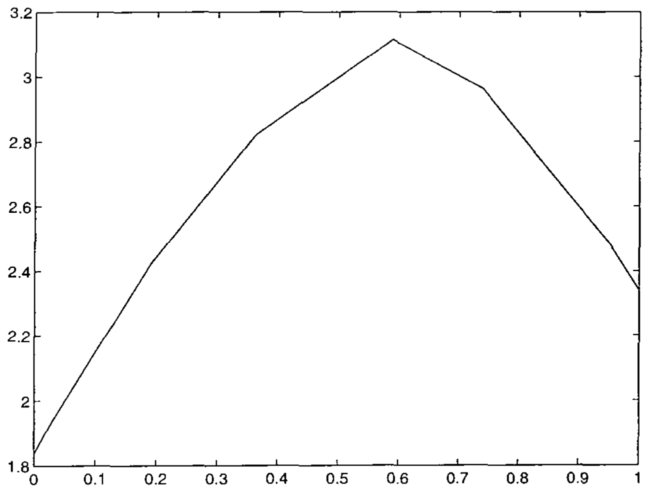


Figure 4.15: Optimal cost function $\overline{V}(p)$ in Class 3 with parameter values as in Example 4.5

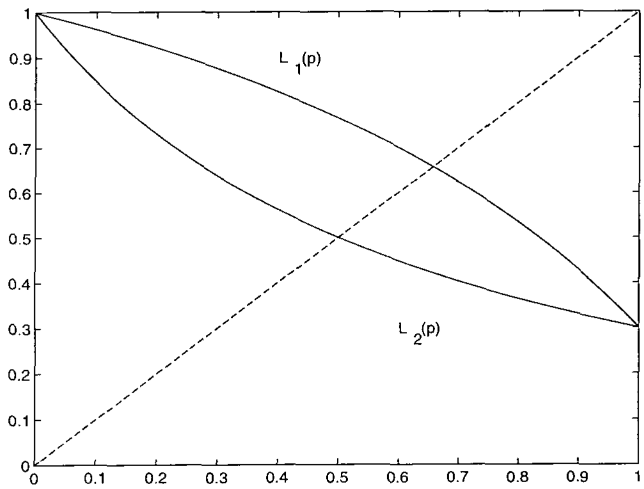


Figure 4.16: $L_1(p)$ and $L_2(p)$ in Class 3 with parameter values as in Example 4.5

∎

4.4.4 Class 4: $P_1 < P^*$ and $P_2 > P^*$

In this case, $L_2(p) > L_1(p)$ and $\Delta > 0$. In order to examine this case, we first need to prove another result about the L_i.

Lemma 4.3

When $\Delta > 0$, L_{21} and L_{12} are increasing in p and $L_{12}(p) > L_{21}(p)$ for $p \in [0,1]$.

Proof *due to Weber [19]*

The proof works by developing the composition inequality which is easily seen to hold for linear functions. Let us note for some parameterisation of the subgradients,

$$L_1(p) = \sup_{s \in [0,1]} \{a_1(s) + b_1(s)p\}$$

and

$$L_2(p) = \inf_{t \in [0,1]} \{a_2(t) + b_2(t)p\}$$

Now

$$L_{21}(p) = L_1(L_2(p)) = \sup_{s \in [0,1]} \inf_{t \in [0,1]} \{a_1(s) + b_1(s)[a_2(t) + b_2(t)p]\}$$

Consider $y_1 = a_1(s) + b_1(s)a_2(t)$. We shall show $y_1 \leq y_2 = a_2(t) + b_2(t)a_1(s)$. Recall that $L_1(0) = L_2(0) = 1 - b$, $L_1(1) = L_2(1) = a$, $L_2(p) \geq L_1(p)$ and L_1 is convex, and L_2 is concave. From these facts (see figure 4.16 below) it follows that $a_1(s) \leq a_2(t)$ and $a_1(s) + b_1(s) \leq \min[1, a_2(t) + b_2(t)]$.

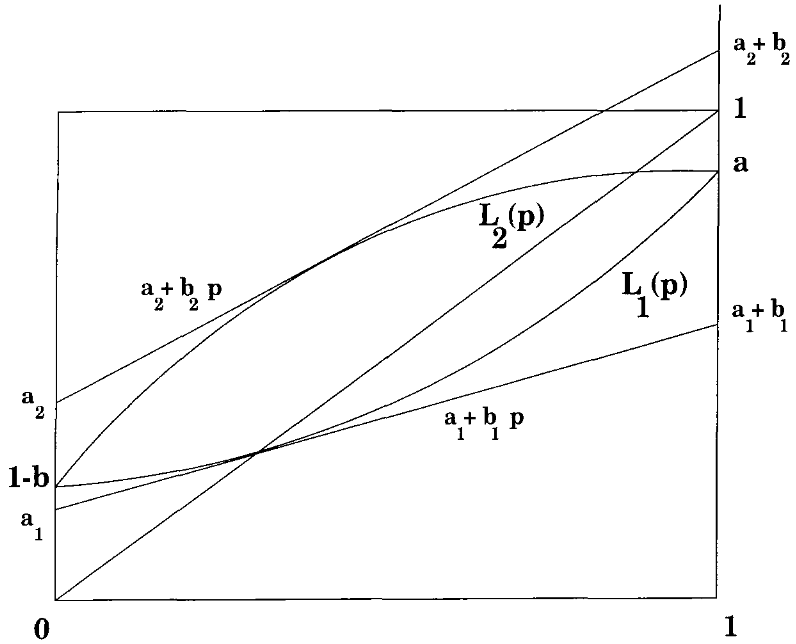


Figure 4.17: Subgradients of L_1 and L_2

So

$$\begin{aligned} y_2 - y_1 &\geq a_2(t) + a_1(s)[a_1(s) + b_1(s) - a_2(t)] - [a_1(s) + b_1(s)a_2(t)] \\ &= (a_2(t) - a_1(s))[1 - a_1(s) - b_1(s)] \\ &\geq 0 \end{aligned}$$

Hence

$$\begin{aligned} L_{21}(p) = L_1(L_2(p)) &= \sup_{s\in[0,1]} \inf_{t\in[0,1]} \{a_1(s) + b_1(s)[a_2(t) + b_2(t)p]\} \\ &\leq \sup_{s\in[0,1]} \inf_{t\in[0,1]} \{a_2(t) + b_2(t)[a_1(s) + b_1(s)p]\} \\ &\leq \inf_{t\in[0,1]} \sup_{s\in[0,1]} \{a_2(t) + b_2(t)[a_1(s) + b_1(s)p]\} \\ &= L_2(L_1(p)) \\ &= L_{12}(p) \end{aligned}$$

∎

From Lemma 4.3 it follows that $L_{12}(p) > L_{21}(p)$ for all $p \in [P_1, P_2]$ and so in particular $L_{12}(P^*) > L_{21}(P^*)$. It is helpful to split this case into three further subcases, one where $L_{21}(P^*) < P^* < L_{12}(P^*)$ and the others where $P^* < L_{21}(P^*)$ or $P^* > L_{12}(P^*)$.

Subclass 4.1: $L_{21}(P^*) < P^* < L_{12}(P^*)$

In this case $U_1(P^*) > L_2(P^*)$ and $U_2(P^*) < L_1(P^*)$ as shown in figure 4.18 below (recall that $L_{12} = L_2 \circ L_1$)

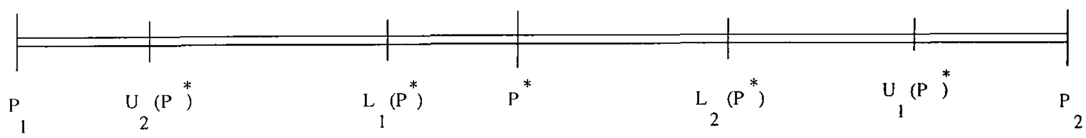


Figure 4.18: Ordering of important points in case 4.1

Neither of Theorems 4.1 or 4.2 apply to this case but it is quite simple in structure. For an initial $p > P^*$ repeatedly search site 1 until the interval $[L_1(P^*), P^*]$ is reached and similarly for $p < P^*$ search in site 2 until the interval $[P^*, L_2(P^*)]$ is reached. From the usual argument using the threshold policy and an induction argument it is clear the strategy $\langle 1, 2\rangle$ is used on $[P^*, U_1(P^*)]$ and strategy $\langle 2, 1\rangle$ is used on $[U_2(P^*), P^*]$. P^* can now be determined in the usual fashion and the optimal policy is clear. $\overline{V}$ has corners at the points

$$Q_2^\ell < \ldots < Q_2^1 < P^* < Q_1^1 < \ldots < Q_1^r$$

(where ℓ and r are chosen as before). The optimal strategy on $[Q_1^k, Q_1^{k+1}]$, for example, is $(\langle 1\rangle_k, \langle 2, 1\rangle)$.

Repeating our procedure with the earlier cases we can find, using for example the

Maple symbolic algebra package:

$$P^* = -\frac{C_1(\alpha_1\alpha_2(1-b-a^2)+\alpha_1 a^2+\alpha_2(a+b-ab)-a(1-b)-1)}{C_1(\alpha_1\alpha_2(2\Delta-b)-a(\alpha_1+\alpha_2)-b+2)+C_2(\alpha_1\alpha_2(2\Delta-a)-b(\alpha_1+\alpha_2)-a+2)} - \frac{C_2(1-b)(\alpha_1\alpha_2(1-b)-\alpha_1 a+\alpha_2 b+a-1)}{C_1(\alpha_1\alpha_2(2\Delta-b)-a(\alpha_1+\alpha_2)-b+2)+C_2(\alpha_1\alpha_2(2\Delta-a)-b(\alpha_1+\alpha_2)-a+2)}$$

In the case where $\alpha_1 = \alpha_2$ and $C_1 = C_2$ we find

$$P^* = \frac{2-b+\alpha(b^2-3b+2-a^2)}{4(1+\alpha)-(a+b)(1+3\alpha)} \qquad (P4)$$

Example 4.6

If we set $a = 0.67$, $b = 075$, $\alpha_1 = 0.19$, $\alpha_2 = 0.22$, $C_1 = 1.3$, $C_2 = 1$ then we find an example of case 4.1, where $P^* = 0.5324$. To check this, we can calculate $P_1 = 0.2787$, $P_2 = 0.6201$, $L_{12}(P^*) = 0.5380$ and $L_{21}(P^*) = 0.3436$. The graphs in this case are shown below.

∎

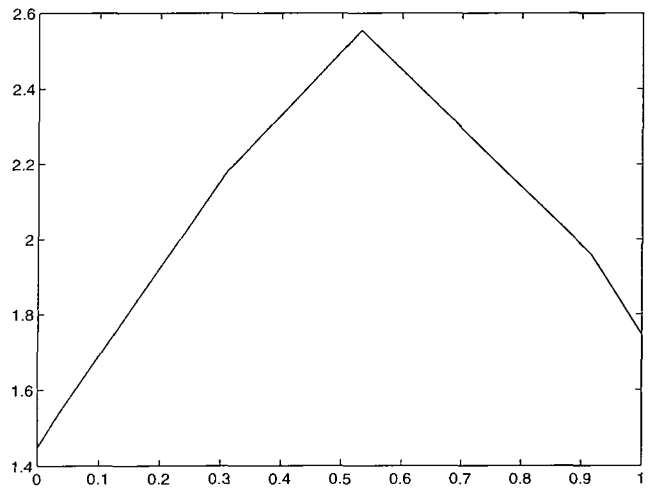


Figure 4.19: Optimal cost function $\overline{V}(p)$ in Class 4.1 for parameter values as in Example 4.6

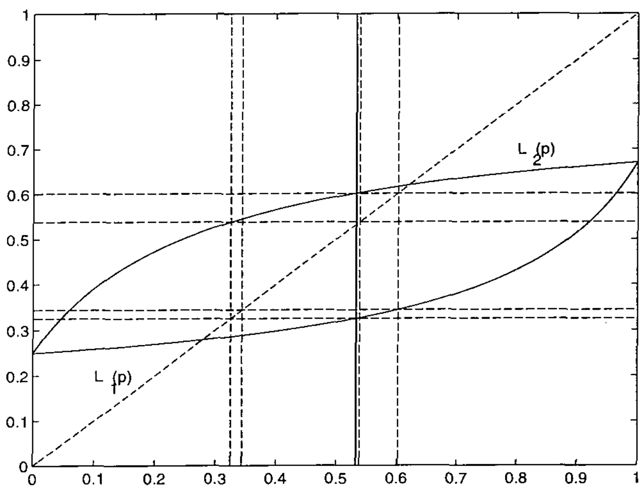


Figure 4.20: $L_1(p)$ and $L_2(p)$ in Class 4.1 for parameter values as in Example 4.6, showing that $L_{12}(P^*) > P^* > L_{21}(P^*)$ - ordering of points is $L_1(P^*)$, $L_{21}(P^*)$, P^*, $L_{12}(P^*)$, $L_2(P^*)$

Subclass 4.2: $P^* < L_{21}(P^*)$ **or** $P^* > L_{12}(P^*)$

These two cases are much the same as case 4.1 in structure though they are rather more complex. The graph below illustrates the first of these cases but as they are similar we consider only the case $L_{12}(P^*) > L_{21}(P^*) > P^*$.

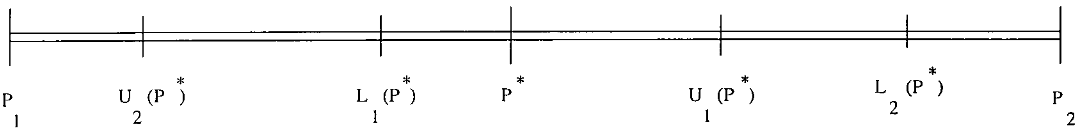


Figure 4.21: Ordering of important points in case 4.2 when $L_{12}(P^*) \geq L_{21}(P^*) \geq P^*$

From the inequalities defining this subcase we have $L_2(P^*) > U_1(P^*) > P^* > L_1(P^*) > U_2(P^*)$. The sequence $\{Q_1^k\}_k$ increases to 1 so let $m \geq 1$ be such that $Q_1^m < L_2(P^*) \leq Q_1^{m+1}$. Let τ denote the string of $m+1$ actions $(2, \langle 1\rangle_m)$. Lemmas 4.1 and 4.2 apply to L_τ which thus has a fixed point P_τ. Since $L_\tau(P^*) > P^*$ (or equivalently $Q_\tau = U_\tau(P^*) < P^*$), $P_\tau > P^*$ and the sequence Q_τ^k decreases to zero as k increases. As $L_{12}(P^*) < L_2(P^*)$ and $L_{12}(P^*) > L_{21}(P^*) > Q_1^{m-1}$ we see that $L_{12}(P^*) \in (Q_1^{m-1}, L_2(P^*))$ and also $L_{\tau 1}(P^*) = L_1^m \circ L_{21}(P^*) < L_1^m \circ L_{12}(P^*) = L_{1\tau}(P^*) \leq P^*$ from which it follows immediately that $L_\tau(P^*) < Q_1$.

If $L_{12}(P^*) \geq Q_1^m$ then $L_{1\tau}(P^*) \geq P^*$ and so $L_\tau : [L_1(P^*), P^*] \mapsto [L_{1\tau}(P^*), L_\tau(P^*)] \subset [P^*, Q_1]$ and $L_1 : [P^*, Q_1] \mapsto [L_1(P^*), P^*]$. Each action in the string $(\tau, 1)$ conforms with the threshold policy which is thus $\langle \tau, 1\rangle$ on $[L_1(P^*), P^*]$. The threshold strategy on $[P^*, Q_1]$ is $\langle 1, \tau\rangle$.

If $L_{12}(P^*) < Q_1^m$ then $L_{1\tau}(P^*) < P^*$ and also $L_1(P^*) < Q_\tau$. Now either $Q_\tau < L_{\tau 1}(P^*)$ or $Q_\tau^{n+1} \leq L_{\tau 1}(P^*) < Q_\tau^n$ for some $n \geq 1$. If $L_{\tau 1}(P^*) \in (Q_\tau^{n+1}, Q_\tau^n]$ for some $n \geq 1$ then $L_{\tau 1} : [Q_\tau, P^*] \mapsto [L_1(P^*), L_{\tau 1}(P^*)] \subset [L_1(P^*), Q_\tau^n]$ and for $j = n, n-1, \ldots, 1$, $L_\tau : [Q_\tau^{j+1}, Q_\tau^j] \mapsto [Q_\tau^j, Q_\tau^{j-1}]$. It follows, after careful tracking, that the threshold strategy on $[Q_\tau, P^*]$ is $\langle \tau, 1, \langle \tau \rangle_n \rangle$ while on $[P^*, L_\tau(P^*)]$ it is $\langle 1, \langle \tau \rangle_{n+1} \rangle$.

It remains to treat the cases where $L_{1\tau}(P^*) < P^*$ and $L_{\tau 1}(P^*) \in (Q_\tau, P^*]$. From $L_{1\tau}(P^*) < P^*$ we know that the sequence $Q^k_{1\tau}$ increases with k and $Q_{1\tau} > P^*$. In addition, as $L_{\tau 1}(P^*) > Q_\tau$ if follows that $L_\tau(P^*) > U_1(Q_\tau) = Q_{1\tau}$. Introduce the strings $\sigma_1 = 1\tau$ and $\sigma_2 = \tau$. We are left with a case where the key inequalities are $L_{\sigma_2}(P^*) > U_{\sigma_1}(P^*) > P^* > L_{\sigma_1}(P^*) > U_{\sigma_2}(P^*)$ which are equivalent to those illustrated in Figure 4.21.

This class thus splits into infinitely many nested subcases in any one of which, in theory, P^* can determined by developing the usual method (the actual expressions rapidly become extremely complicated). The corners of $\overline{V}$ can be found by working outwards as before.

Example 4.7

Consider $a = 0.95$, $b = 0.6$, $\alpha_1 = 0.19$, $\alpha_2 = 0.22$, $C_1 = 1.3$, $C_2 = 1$. We find that $P^* = 0.6433$ with $L_{12}(P^*) = 0.8632$ and $L_{21}(P^*) = 0.7333$. Hence we are in Subclass 4.2 as $P^* < L_{21}(P^*)$. Graphically, we have Figures 4.22 and 4.23 below. An analagous example where $P^* > L_{12}(P^*)$ can be found by swapping the parameters site-wise i.e., when $a = 0.6$, $b = 0.95$, $\alpha_1 = 0.22$, $\alpha_2 = 0.19$, $C_1 = 1$ and $C_2 = 1.3$. Under these circumstances, we find $P^* = 0.3566$ with $P_1 = 0.0573$, $P_2 = 0.5171$, $L_{12}(P^*) = 0.2663$ and $L_{21}(P^*) = 0.1366$.

∎

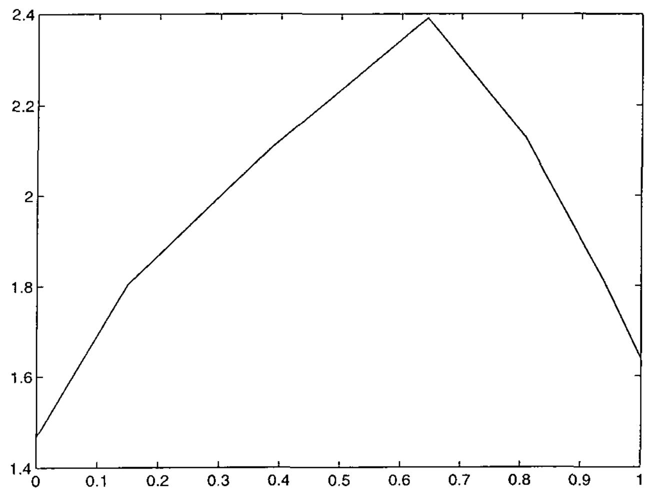


Figure 4.22: Optimal cost function $\overline{V}(p)$ in Class 4.2 for parameter values as in Example 4.7

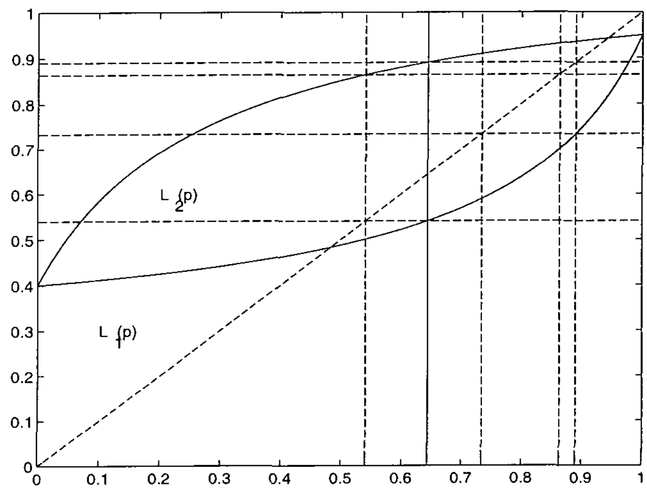


Figure 4.23: $L_1(p)$ and $L_2(p)$ in Class 4.2 for parameter values as in Example 4.7, showing that $L_{12}(P^*) > L_{21}(P^*) > P^*$ - ordering of points is $L_1(P^*)$, P^*, $L_{21}(P^*)$, $L_{12}(P^*)$, $L_2(P^*)$

4.5 Verification of Case Boundaries.

In section 4.4 we divided the problem of checking the optimality of a threshold policy into cases and found expressions for P^* in each of these cases. The calculated $\overline{V}$ is the unique solution to the optimality equation (4.4). It remains to check that all values of a, b, the α_i and the C_i result in a problem that falls into one of the listed cases. Let P_i^* denote the value of P^* found for case i.

Consider the region $\Delta \leq 0$ or equivalently $a + b \leq 1$. In this region $P_1 \geq P_2$ and case 4 cannot occur. Case 1 is described by the condition $P_1^* \leq P_2 \leq P_1$ while case 3 is described by the condition $P_2 \leq P_3^* \leq P_1$. These cases have a common boundary along a curve upon which $P_1^* = P_3^* = P_2$. To check this we used a symbolic algebra package to solve the equation $P_1^* = P_2$ for b in terms of a, the α_i and the C_i and then repeated this for the equation $P_3^* = P_2$. We found that b was the real root in $[0, 1]$ of the same cubic in each case. The actual expression is very long but in the special case where $\alpha_i = \alpha$ and the $C_i = C$ it simplifies to

$$\alpha(1+\alpha)b^3 + (\alpha(1+\alpha)a - 2\alpha(3+\alpha))b^2 + ((1-\alpha)^2 a - \alpha(1+\alpha)a^2 + 8\alpha)b$$
$$+ 1 + a^2 - 3a - \alpha(a^3 - 4a^2 + 2a + 1) - \alpha^2(a-1)^3 = 0 .$$

Further details of this calculation can be found in the appendices. The boundary between cases 2 and 3 can be found in a similar fashion (in fact the expression found for this boundary is the same as given above but with a and b exchanged). This establishes that the piecewise linear function $\overline{V}$ constructed in section 4.4 is a solution of the optimality equation (4.4) for all α_i and C_i and a, b such that $a + b \leq 1$.

When $a + b > 1$ things are less straightforward. This time $P_1 < P_2$ and case 3 cannot occur. The boundary between case 1 and case 4 occurs along a curve where $P_1^* = P_1$ and that between cases 2 and 4 along a curve where $P_2^* = P_2$. The boundaries of case 4.1 are along the curves where $P_4^* = P_{21}$ (the fixed point of L_{21}) and $P_4^* = P_{12}$.

Figure 4.24 below shows which class of problem is obtained for all values of a and b

in $(0,1)$ in the case where $C_i = C$ and $\alpha_i = \alpha$ for $\alpha = 0.2$.

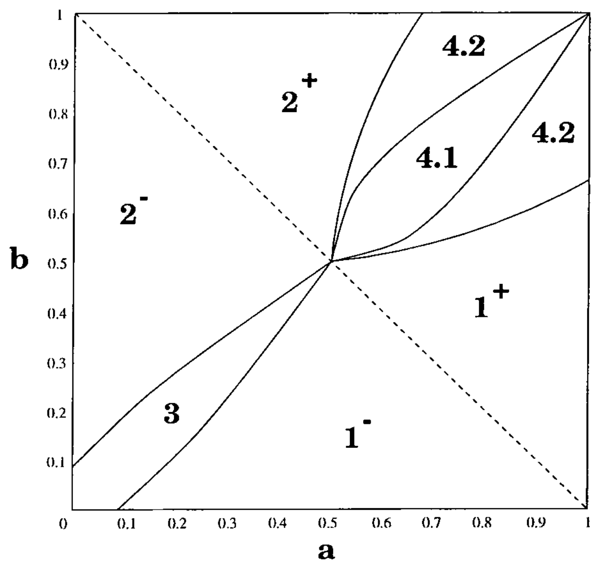


Figure 4.24: Regions of optimal P^*

On the boundary between cases 1 and 4 $P_1^* = P_1$. Recall that the optimal strategy in case 1 is $\langle 1 \rangle$ on $[P^*, 1]$ and $(2, \langle 1 \rangle)$ on $[Q_2^1, P^*]$. The other side of the boundary corresponds to case 4.2 as $m \to \infty$ where formally the optimal strategy on a neighbourhood below P^* is $(2, \langle 1 \rangle)$ while on a neighbourhood above P^* it is $(1, 2, \langle 1 \rangle)$ which seems reasonable.

The form of the piecewise linear $\overline{V}$ constructed in section 3 together with the calculations of boundaries above have the following result as an immediate consequence.

Theorem 4.3

A threshold strategy is optimal for all parameter values α_i, C_i and (a, b) in regions 1, 2, 3, 4.1 and the various cases of 4.2 for which a threshold strategy can be determined. In particular this includes all a and b such that $a + b \leq 1$.

∎

Numerical and computer based symbolic investigations support the conjecture that there are no values of (a, b) in region 4.2 for which Ross's conjecture fails (these would manifest themselves as gaps between the boundaries of the various subcases) but the author cannot prove this. Methods examined include straightforward iteration of a discretisation of all possible values of a and b while keeping α_i and C_i fixed. However, as can be seen from the above arguments, the interpretation given is a coherent one, and one that appears to be correct. However, a proof remains elusive.

4.6 Conclusion

In this chapter, we have studied in some detail the simplest case scenario for searching for a moving target. It is clear that such problems are far from simple when considered in a rigorous mathematical format, and I hope to have demonstrated in the course of the chapter how complex the solutions can be, and offered some explanation as to why these questions have remained unanswered for so long. We now go on to look at extensions of this problem, where the solutions are often unattainable.

Chapter 5

Extensions and Related Problems

The outcome of any serious research can only be to make two questions grow where only one grew before.

Thorstein Veblen – The Place of Science in Modern Civilisation and Other Essays

5.1 Introduction

This chapter is concerned with problems similar in nature to the two site search problem examined in Chapter 4. Each is tangentially related but poses problems which cannot be adequately solved as yet. Where possible, we conjecture results suggested by computer simulation and extensions of 2 site theory.

The first problem we look at is the 3 site search problem - i.e., a direct extension of the 2 site problem with an extra site.

5.2 A Three Site Search Problem

In this case, we have as our state variable the probability triple

$$p = (p_1, p_2, p_3)$$

The target follows a path generated by the transition probability matrix M where

$$M = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix}$$

Overlook probabilities and costs are as previously defined. From these, we can extend the definition of $L_i(p)$ as in (3.3) to find

$$L_1(p) = \left[\frac{\alpha_1 M_{11}p_1 + M_{21}p_2 + M_{31}p_3}{1-(1-\alpha_1)p_1}, \frac{\alpha_1 M_{12}p_1 + M_{22}p_2 + M_{32}p_3}{1-(1-\alpha_1)p_1}, \frac{\alpha_1 M_{13}p_1 + M_{23}p_2 + M_{33}p_3}{1-(1-\alpha_1)p_1}\right] \tag{5.1}$$

$$L_2(p) = \left[\frac{M_{11}p_1 + \alpha_2 M_{21}p_2 + M_{31}p_3}{1-(1-\alpha_2)p_2}, \frac{M_{12}p_1 + \alpha_2 M_{22}p_2 + M_{32}p_3}{1-(1-\alpha_2)p_2}, \frac{M_{13}p_1 + \alpha_2 M_{23}p_2 + M_{33}p_3}{1-(1-\alpha_2)p_2}\right] \tag{5.2}$$

$$L_3(p) = \left[\frac{M_{11}p_1 + M_{21}p_2 + \alpha_3 M_{31}p_3}{1-(1-\alpha_3)p_3}, \frac{M_{12}p_1 + M_{22}p_2 + \alpha_3 M_{32}p_3}{1-(1-\alpha_3)p_3}, \frac{M_{13}p_1 + M_{23}p_2 + \alpha_3 M_{33}p_3}{1-(1-\alpha_3)p_3}\right] \tag{5.3}$$

and we find that our optimal cost function $V(p)$ must satisfy

$$\begin{aligned} V(p) &= \min\{C_1 + V(L_1(p))(1-(1-\alpha_1)p_1);\ C_2 + V(L_2(p))(1-(1-\alpha_2)p_2); \\ &\qquad C_3 + V(L_3(p))(1-(1-\alpha_3)p_3)\} \end{aligned} \tag{5.4}$$

We can immediately see that in this problem we are dealing with a far more complicated set of equations than previously. This is reflected in the fact that only one paper has ever been published on the topic, by Nakai [8]. His paper is concerned only with perfect detection and only examined highly constrained transition probability matrices, such as entirely symmetric matrices of the form

$$M = \begin{pmatrix} a & \frac{1-a}{2} & \frac{1-a}{2} \\ \frac{1-a}{2} & a & \frac{1-a}{2} \\ \frac{1-a}{2} & \frac{1-a}{2} & a \end{pmatrix}.$$

As a result it is difficult to extend his work at all, as his methods are very much tailored to such cases.

When we look at such problems, we need to consider what it is that we are trying to achieve. From the results of Chapter 2, we can conjecture the following

Conjecture 5.1

In a general 3-site case, the optimal first look strategy regions $\mathcal{A}_i$ are star convex with respect to the corner points e_i, $i = 1, 2, 3$.

∎

However it is seemingly not possible to prove it by the methods shown in Chapter 2, although we can prove its validity under certain constraints (i.e. $\alpha_i = \alpha$ and $C_i = C\ \forall i$), by using Corollary 3.1 and Theorem 3.2. It is our aim to determine exactly what these regions $\mathcal{A}_i$ actually are in any given case. In a 2-site case this was a simple procedure, as the regions were immediately characterised by the knowledge of the point P^*. In the 3-site case and beyond however life is not so straightforward, as knowledge of P^* (in this case the point where all three regions meet) does not give us enough useful information. What we in fact need to do is calculate the 3 boundary lines between different regions as highlighted in Figure 5.1 below.

Unfortunately, none of the techniques used in the 2-site scenario as discussed in Chap-

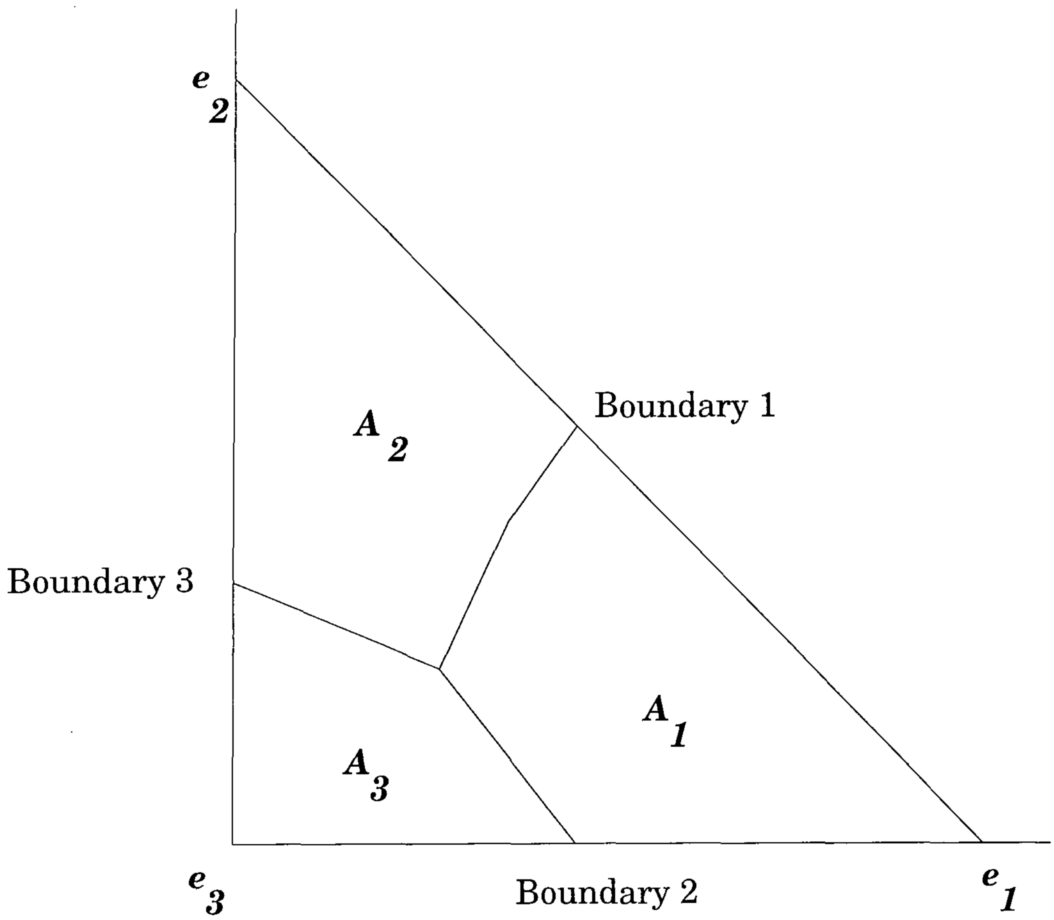


Figure 5.1: Boundary Lines in 3-site case

ter 4 apply adequately to such a problem, as we cannot merely concern ourselves with the optimal policies near P^*. In fact, we need to know the optimal policies right the way along the boundaries in order to calculate them. As can be imagined, this might involve a huge number of different regions which it would be necessary to calculate. Correspondingly, there are going to be a vast number of different boundaries, each of which would apply to some group of transition probabilities, search costs and overlooks. It seems unlikely that the divisions would be as clean as the were in the 2-site case. It seems likely that the formulae for boundaries between regions will vary depending on the exact parameters, and moreover are likely to be very complicated and unwieldy to use or implement. Hence, such an approach seems difficult, if not infeasible. We can conjecture further however the following.

Conjecture 5.2

In a general 3-site search problem, the optimal cost function V is piecewise linear i.e. the unit simplex in three dimensions can be divided into regions R such that $V(p) = \sum_{i=1}^{3} s_i p_i + k$ for $p \in R$, where the s_i and k are R-dependent constants.

∎

This concept is strengthened by the extension of Example 4.1 to 3 sites. More importantly, numerical approximations to the optimal cost function V using the Matlab program **leprechaun2.m** found in Appendix A, bear these conjectures out. Consider the following example.

Example 5.1

If we set

$$M = \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.9 & 0.01 & 0.09 \\ 0.95 & 0.03 & 0.02 \end{pmatrix}$$

and set $\alpha_i = 0.4$, $\forall i$ and $C_1 = 1$, $C_2 = C_3 = 0.5$ then the approximate optimal cost function and optimal first choice regions are given in figure 5.2, 5.3 and 5.4 overleaf.

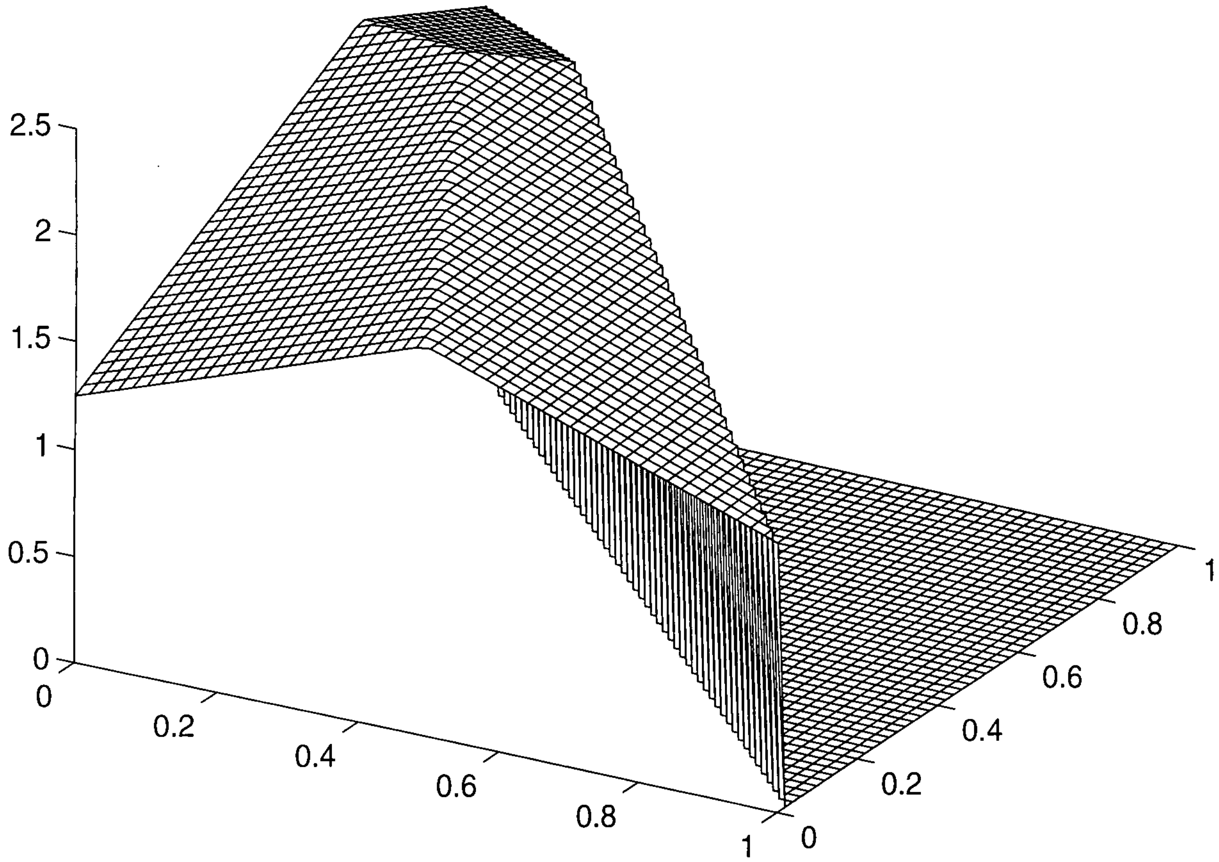


Figure 5.2: Optimal Cost Function V for Parameter values as in Example 5.1

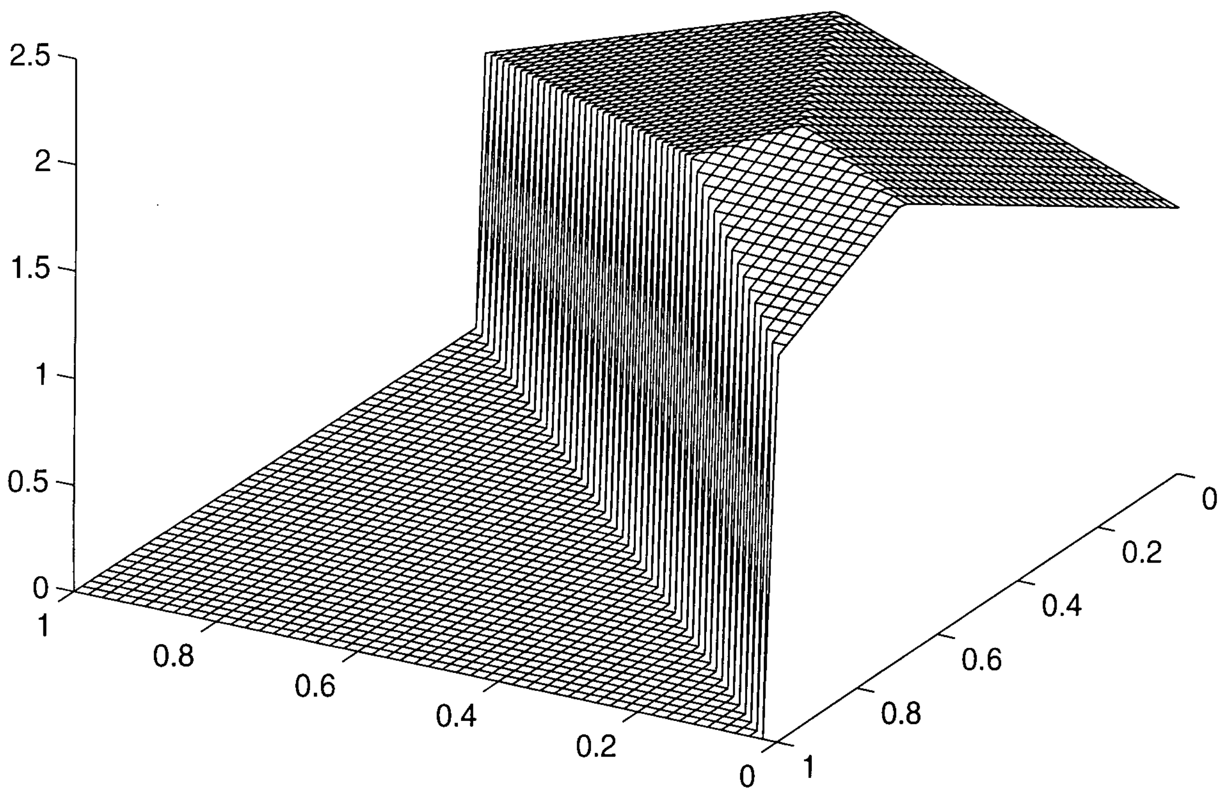


Figure 5.3: Reverse view of Figure 5.2

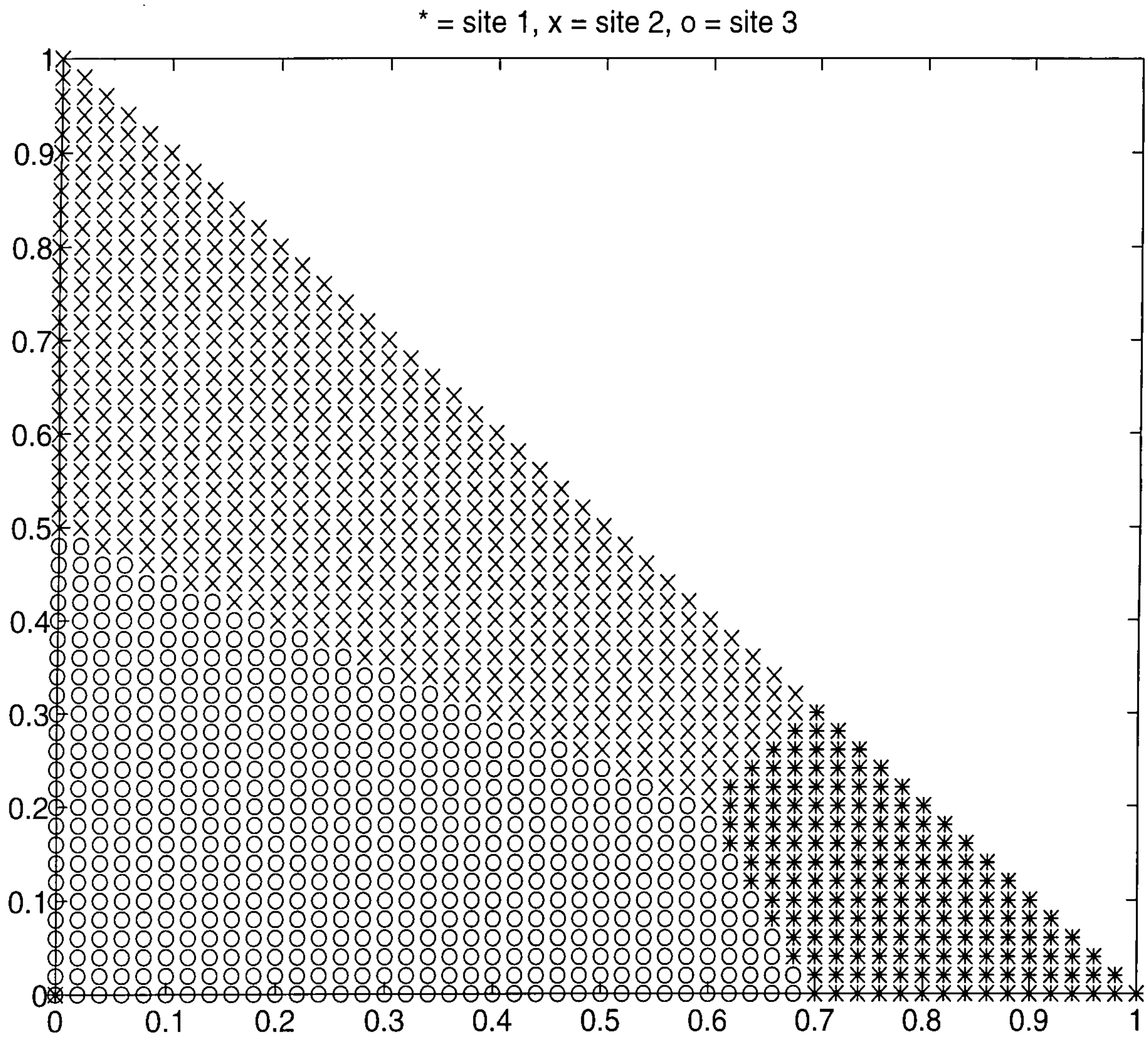


Figure 5.4: Optimal first choice regions for parameters as in Example 5.1

However, we can say quite a lot about such an example. Based on the results of chapter 4, we would expect such a set of parameters to take us to a region where the policy $\{always\ look\ in\ site\ 1\}$ would be optimal. If we assume piecewise linearity throughout, and consider such a policy, we can find the equation of the plane determined by such an action exactly. In fact, if we define

$$V(p) = A_1 p_1 + A_2 p_2 + A_3$$

for $p \in \mathcal{A}_1$, then we find, by substitution into 5.4 above

$$A_1 p_1 + A_2 p_2 + A_3 = C_1 + V(L_1(p))(1 - (1-\alpha)p_1)$$

which we can solve exactly to get

$$A_1 = \frac{C_1(1 - M_{22} - M_{32})}{M_{31}(M_{22} - 1) - M_{21}M_{32}}$$

$$A_2 = \frac{C_1(M_{21} - M_{31})}{M_{31}(M_{22} - 1) - M_{21}M_{32}}$$

$$A_3 = \frac{C_1[(M_{22} - 1)(M_{31} + 1) - M_{32}(1 + M_{21}) + \alpha(M_{11}(1 + M_{32} - M_{22}) + M_{12}(M_{21} - M_{31}))]}{(1-\alpha)(M_{31}(M_{22} - 1) - M_{21}M_{32})}$$

Hence, we know the exact equation of a plane for $V(p; \langle 1 \rangle)$. We can compare this to the approximate V calculated, getting the following graphs, when first overlaying one graph on the other and then subtracting one from the other.

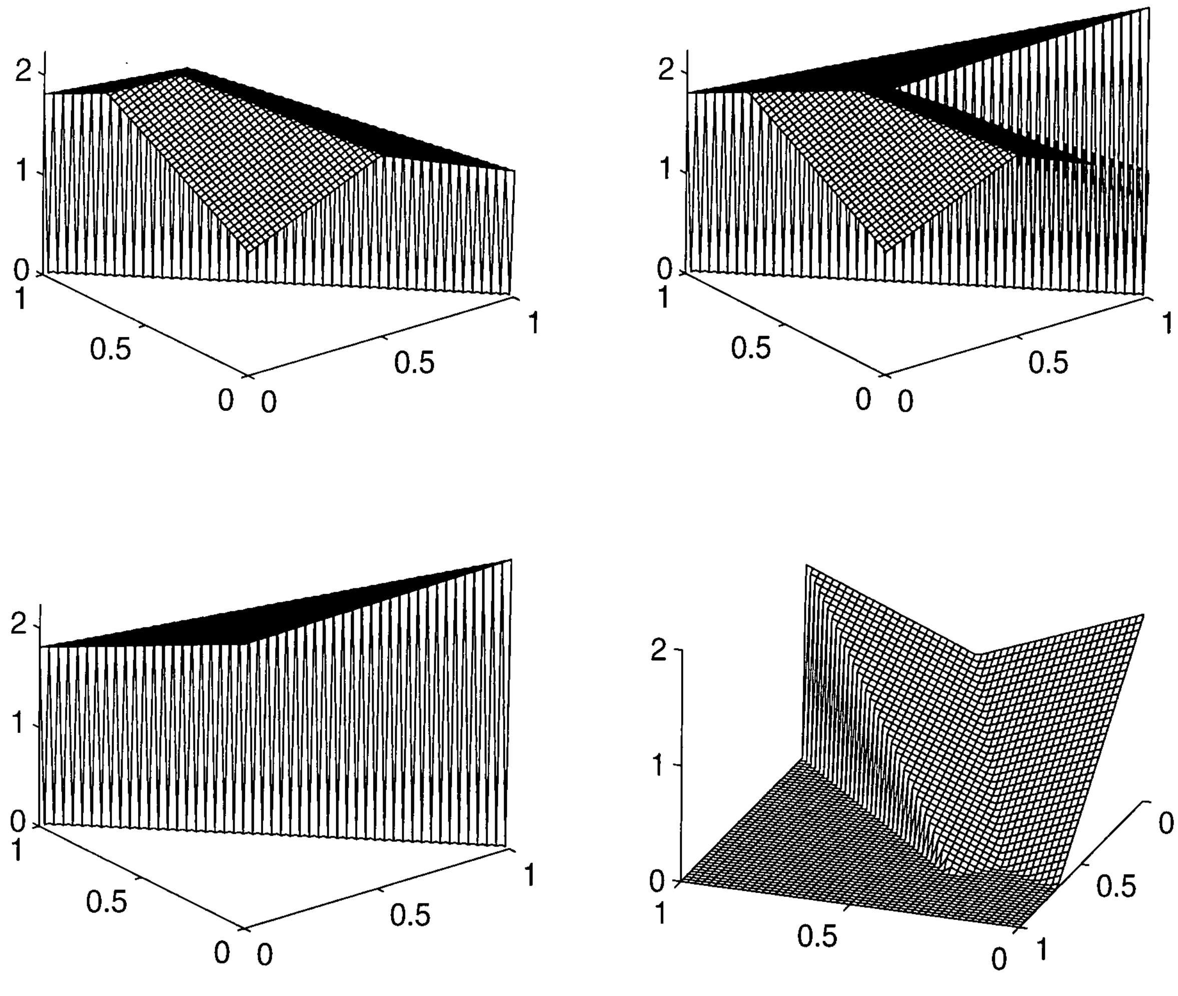


Figure 5.5: Overlaying Exact Plane on Approximate Cost Function

We can check that the optimal policy is indeed of the form $(a_1, \langle 1 \rangle)$ with $a_1 = i$ on $\mathcal{A}_i$, by examining where the $L_i(p)$ take us. Figure 5.6 shows clearly that we lie entirely in $\mathcal{A}_1$ after one look, no matter which state we start from.

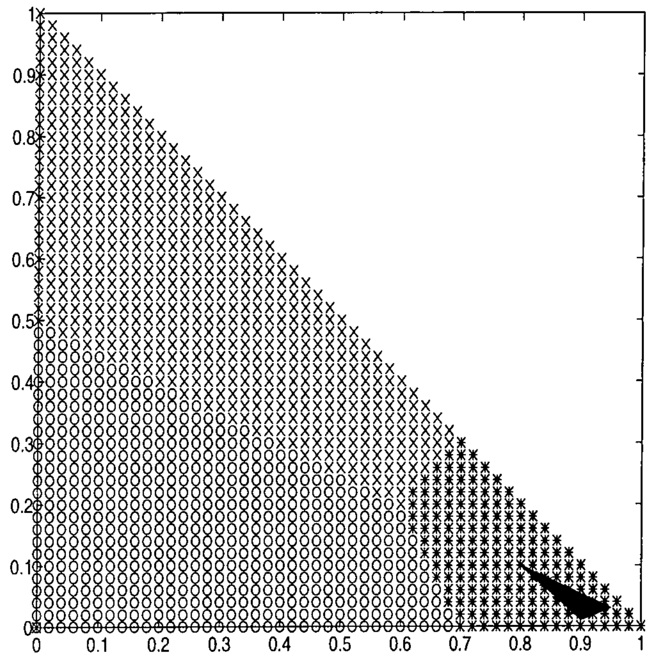


Figure 5.6: Shaded area shows updated states for parameters as in Example 5.1

Hence we have a three region optimal policy, with $(\langle 1 \rangle)$ being optimal for $p \in \mathcal{A}_1$, $(2, \langle 1 \rangle)$ optimal for $p \in \mathcal{A}_2$ and $(3, \langle 1 \rangle)$ optimal for $p \in \mathcal{A}_3$. The equations of the other two planes can be found in a similar fashion as before, and they are given by (using Maple output, where C_i is denoted by K_i, and α is given by a).

```
solve(A1=A1*a*M11-A1*M31+A2*a*M12-A2*M32-A3*(1-a), A2=A1*M21-
A1*M31+A2*M22-A2*M32, A3=K1+A1*M31+A2*M32+A3, B1=A1*M11-
A1*M31+A2*M12-A2*M32, B2=a*A1*M21-A1*M31+a*A2*M22-A2*M32-
A3*(1-a), B3=K2+A1*M31+A2*M32+A3, C1=A1*M11-a*A1*M31+A2*M12-
a*A2*M32+A3*(1-a), C2=A1*M21-a*A1*M31+A2*M22-a*M32*A2+A3*(1-
a), C3=K3+A1*a*M31+A2*a*M32+A3*a, C1,C2,C3,A1,A2,A3,B1,B2,B3);
```

$$\Big\{ B3 = -(-M21\,M32\,K2 - M31\,M22\,a\,K2 + M31\,M22\,a\,K1$$

$$- M21\,M32\,a\,K1 - M31\,K2 + a\,M12\,M21\,K1 - M32\,K1$$

$$- M31\,a\,K1 - M31\,a\,M12\,K1 + M31\,M22\,K2 + M31\,a\,K2 - K1$$

$$+ M22\,K1 + a\,M11\,K1 + a\,M11\,M32\,K1 + M21\,M32\,a\,K2$$

$$- M22\,a\,M11\,K1) \big/ (\%2), B2 = -K1(-1 + a\,M11 - a\,M21 + M22$$

$$- M32 - M22\,a\,M11 - a\,M21\,M32 + a\,M12\,M21 + a\,M11\,M32$$

$$+ M31\,a\,M22 - M31\,a\,M12) \big/ (\%1), B1 = K1(-M31 - M22\,M11$$

$$- M31\,M12 + M31\,M22 - M21\,M32 + M12\,M21 + M11$$

$$+ M11\,M32) \big/ (\%1), A2 = -\frac{K1\,(-M21 + M31)}{\%1}, A3 = -K1(-1$$

$$- M31 + a\,M11 + M22 - M21\,M32 + M31\,M22 - M32$$

$$- M22\,a\,M11 + a\,M12\,M21 + a\,M11\,M32 - M31\,a\,M12) \big/ (\%2),$$

$$A1 = \frac{K1\,(-M22 + 1 + M32)}{\%1}, C1 = K1(-M31\,a - M22\,a\,M11$$

$$- a\,M21\,M32 + a\,M11 - M31\,a\,M12 + M31\,a\,M22 + a\,M12\,M21$$

$$+ a\,M11\,M32 + M11 - M31 + M22 - 1 - M22\,M11 - M21\,M32$$

$$+ M12\,M21 + M11\,M32 + M31\,M22 - M31\,M12 - M32) \big/ (\%1),$$

$$C3 = -(-M21\,M32\,K3 - M31\,M22\,a\,K3 + M31\,M22\,K3$$

$$- M31\,a^2\,K1 - M22\,a^2\,M11\,K1 + M31\,a\,K3 - a\,M32\,K1$$

$$- M31\, a^2\, M12\, K1 + M22\, a\, K1 - M31\, K3 + M31\, M22\, a^2\, K1 - a\, K1$$

$$+ a^2\, M11\, K1 + a^2\, M12\, M21\, K1 + M21\, M32\, a\, K3$$

$$+ a^2\, M11\, M32\, K1 - M21\, M32\, a^2\, K1) \Big/ (\%2), C2 = K1 (M21$$

$$- M31 + M22 - 1 - M32 - M31\, a + a\, M11 - M22\, a\, M11$$

$$- a\, M21\, M32 + a\, M12\, M21 + a\, M11\, M32 - M31\, a\, M12$$

$$+ M31\, a\, M22) \Big/ (\%1) \Big\}$$

$$\%1 := -M31 + M31\, M22 - M21\, M32$$

$$\%2 := M31\, a\, M22 + M21\, M32 - M31\, a - M31\, M22 - a\, M21\, M32 + M31$$

Checking these off against the approximate optimal cost function as before we find figure 5.7 and 5.8. This clearly gives evidence that the optimal cost function is indeed piecewise linear in this case.

∎

Of course, not all parameter values are going to work out this easily - this example was chosen to make the calculations and policies easy to predict. By reference to chapter 4, we would expect the policies to be most complicated when M_{11}, M_{22} and M_{33} were all near 1. However, a full proof remains open for this problem, and as yet we can only conjecture as to what the true solution might be.

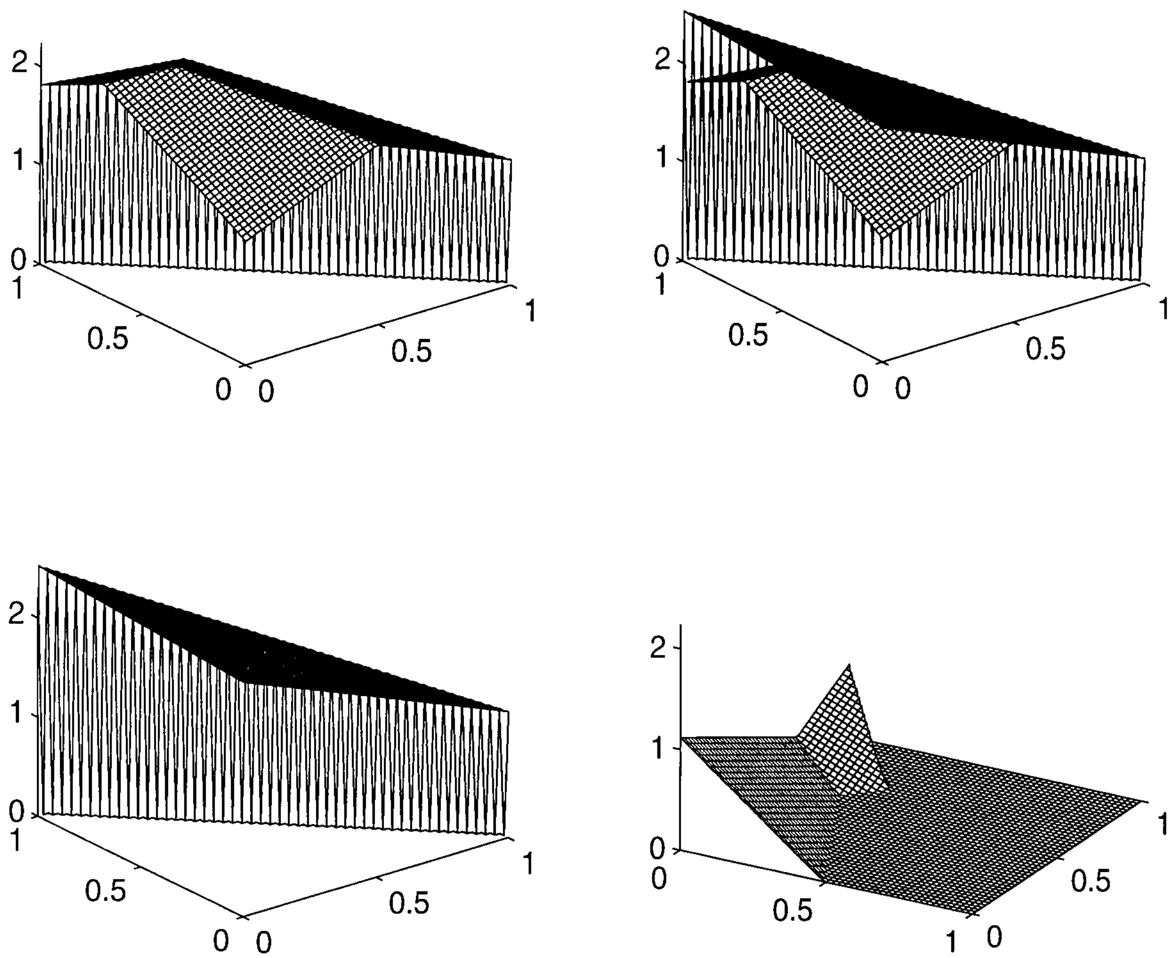


Figure 5.7: Overlaying Exact Plane for policy 2, $\langle 1 \rangle$ on Approximate Cost Function

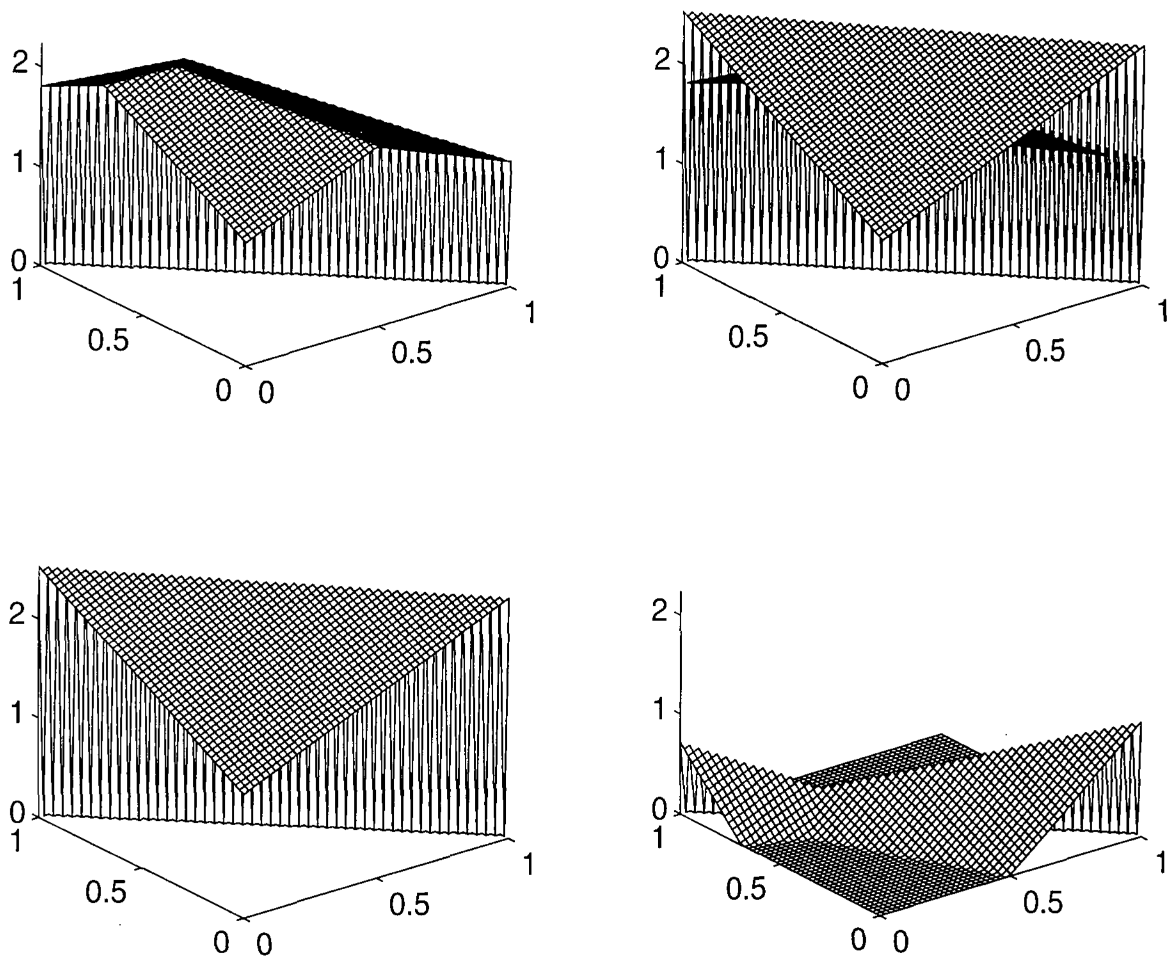


Figure 5.8: Overlaying Exact Plane for policy 3, $\langle 1 \rangle$ on Approximate Cost Function

5.3 Variable Resource Search Problems

We now turn our attention to a different extension of the 2-site search problem discussed in chapter 4 – a search problem on 2 sites where we have a number of possible options at each stage. At each point in time we can divide search effort into a certain number of levels (which may be infinite) and we can search a number of different sites at the same time. This changes the focus of the search problem in that Ross' Conjecture no longer has a meaning, as we would expect there to be a number of intervals over $[0, 1]$ on which different first-step policies were optimal. This problem stems from one first developed by Nakai [9] in 1980. In his paper, he extends Pollock's [11] paper of 1970 to a problem of search over 2 sites with perfect detection and unit cost search. His model allows three choices at each stage - look in site 1, look in site 2, or wait. He gives a complete solution to the problem but offers little room for extension to variable overlook/cost structures. However, he does give interesting results in this case which suggest a general optimal solution format. The general problem in discrete form has never been studied in published form since then. The analogous continuous time version of this problem has been discussed and solved in 1994 by Assaf and Sharlin-Bilitzky [2], in whose paper the target moves in accordance with some continuous time Markov process and there is an infinite division of search resource. Essentially, however, this paper is an extension of that of Weber [18], although the results suggested in it are interesting for comparative purposes. Both [2] and [9] offer a number of different scenarios into which any given case can fall depending on the variable parameters, each of which has a different optimal policy structure. The problem we seek to develop here is a new extension but is clearly related to both of the above papers and the results of those papers must have influence on it.

The problem we discuss is one where we search over 2 sites but with a fixed constant amount of total search resource available to us at any stage. We can divide this search resource up into discrete amounts and apply them as best we see fit. This makes sense as if, for example, we are searching for a liferaft and we have 2 helicopters at our disposal, we have 6 choices at each stage.

(i) both helicopters grounded

(ii) helicopter 1 search site 1, helicopter 2 grounded

(iii) helicopter 1 search site 2, helicopter 2 grounded

(iv) both helicopter 1 and helicopter 2 search site 1

(v) both helicopter 1 and helicopter 2 search site 2

(vi) helicopter 1 search site 1, helicopter 2 search site 2

Obviously costs and overlook probabilities are directly related to the amount of search effort put in. In the model proposed, the costs are linear and the overlook probabilities are proportional to the amount of search effort invested .There is no real justification for these relationships, and indeed other overlook and cost structures would lead to numerically similar results to those shown. Formally, the model is as follows.

We search for a target which can be in one of 2 sites S_1, S_2 and which chooses its next site according to some Markov chain with transition probability matrix M. We have a fixed amount of search resource which can be allocated to search regions S_1, S_2 in any way we see fit. Costs are linear in search allocation as are overlook probabilities. What we say is that it costs us a certain amount C_0 to have our entire search resource idle, and that to search sites S_1 and S_2 at full intensity costs C_1 and C_2 respectively. Suppose we allocate fraction r_1 of our search resource to search site S_1 and r_2 to search S_2. Then this will cost us $r_1C_1 + r_2C_2 + (1 - r_1 - r_2)C_0$. We want our overlook probability for site S_1 to vary from 1 at $r_1 = 0$ to α_1 at $r_1 = 1$, where α_1 is the overlook probability associated with a dedicated search of S_1 as in Chapter 4. Our overlook probability is calculated as $\alpha_1^{r_1}$. Once again, our aim is to try and minimise time to detection.

Once again, let us denote the state of our process by $p = P(\text{target in site } 1)$. At each stage we have a number of choices about how to allocate resources. To give some definitions, if we have n units of resource, and we allocate β_1 of them to site S_1 and β_2 to site S_2 (so $r_1 = \frac{\beta_1}{n}$), then we can update our state after each unsuccessful search attempt using Bayes' theorem to $L_{(\beta_1,\beta_2)}$, where

$$L_{(\beta_1,\beta_2)}(p) = \frac{a(1-\alpha_1^{\frac{\beta_1}{n}})p + (1-b)(1-\alpha_2^{\frac{\beta_2}{n}})(1-p)}{(1-\alpha_1^{\frac{\beta_1}{n}})p + (1-\alpha_2^{\frac{\beta_2}{n}})(1-p)}$$

and our optimality equation is

$$\begin{aligned} V(p) &= \min_{(\beta_1,\beta_2)} \Big\{ \frac{\beta_1}{n}C_1 + \frac{\beta_2}{n}C_2 + (1 - \frac{\beta_1}{n} - \frac{\beta_2}{n})C_0 \\ &\quad + V(L_{(\beta_1,\beta_2)}(p))(1-\alpha_1^{\frac{\beta_1}{n}})p + (1-\alpha_2^{\frac{\beta_2}{n}})(1-p) \Big\} \end{aligned} \tag{5.5}$$

Immediately we can see that this optimality equation is far more complicated than any seen previously. When searching over two sites, with $n = 1$ (the simplest case) we find we have 3 choices at each stage - look in S_1, look in S_2 or do nothing. These have updating formulae

$$L_{(1,0)}(p) = \frac{a\alpha_1 p + (1-b)(1-p)}{\alpha_1 p + (1-p)}$$

$$L_{(0,1)}(p) = \frac{ap + \alpha_2(1-b)(1-p)}{p + \alpha_2(1-p)}$$

$$L_{(0,0)}(p) = ap + (1-b)(1-p)$$

$$\begin{aligned} V(p) &= \min\Big\{C_1 + V(L_{(1,0)}(p))(\alpha_1 p + (1-p))\,; \\ &\qquad C_2 + V(L_{(0,1)}(p))(p + \alpha_2(1-p))\,; \\ &\qquad C_0 + V(L_{(0,0)}(p))\Big\} \end{aligned}$$

which seems marginally more complicated than the optimality equation (4.4). What, however, can we say about the solution of such an optimality equation? Piecewise linearity is suggested and motivated by direct extension of Lemma 4.1 to this case, so it appears likely that the following is true.

Conjecture 5.3

The optimal cost function solving equation 5.5 above is piecewise linear.

∎

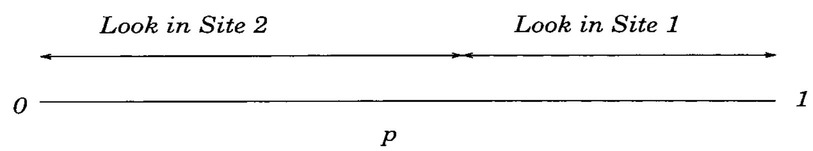
Moreover, the following extension to Ross' Conjecture seems plausible.

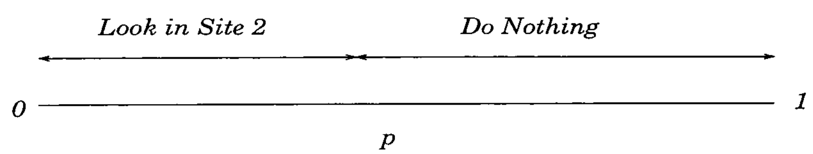
Conjecture 5.4 *(after Ross)*

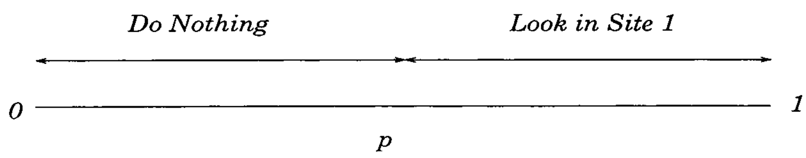
In a 2 site varying resource search problem, β_1 increases monotonely and β_2 decreases monotonely as p increases.

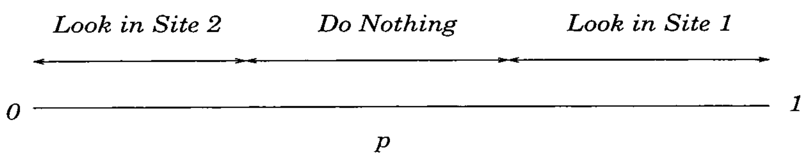
∎

What this means is that when we are fairly certain about the location of the target, we put a lot of effort into searching for it. When we not so certain, we either hedge both ways or we put little effort into searching anywhere. In the $n = 1$ case, it means that we conjecture that we can have the following scenarios.









We can use computer simulation to approximate the optimal cost function, which bear out both of these conjectures, as can be seen from the following example which was produced using the program **resources.m** which can be found in Appendix A. Examples of the four above possible scenarios have been found, and no case has been discovered where one of the structures does not apply. It is worth noting at this point that in their paper covering the analogous continuous time problem, Assaf and Sharlin-Bilitzky [2] found strategies which involved up to 5 regions of optimal first-step decision, but although I have investigated numerous collections of parameter values for the discrete problem, I have found no evidence to suggest that such policies exist in this case.

Example 5.2

Consider

$$\begin{pmatrix} 0.4 & 0.6 \\ 0.2 & 0.8 \end{pmatrix}$$

with $C_1 = 2$, $C_2 = 3$, $C_n = 0.01$ and $\alpha_1 = 0.6$ and $\alpha_2 = 0.7$. We can compare the effect that adding different amounts of resource has to the solution to such a problem. With no option but to search, we are in a normal 2-site search scenario, and our optimal solution looks like Figure 5.9 below. With 2 levels of search resource it is in 5.10, and with 4 levels of search resource, we have Figure 5.11.

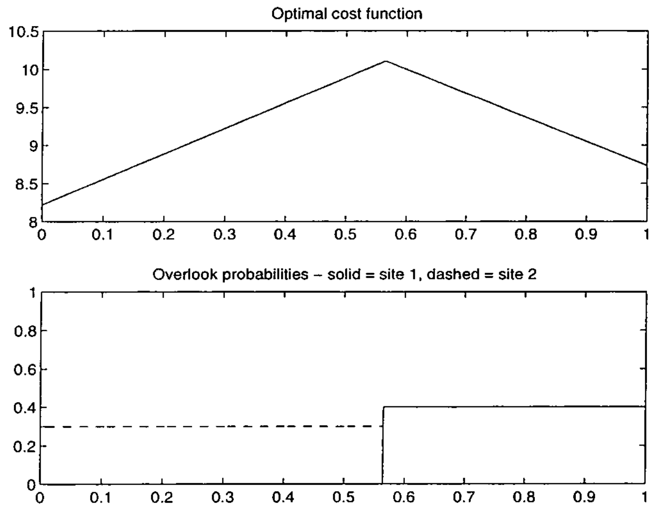


Figure 5.9: Optimal cost function with search options only

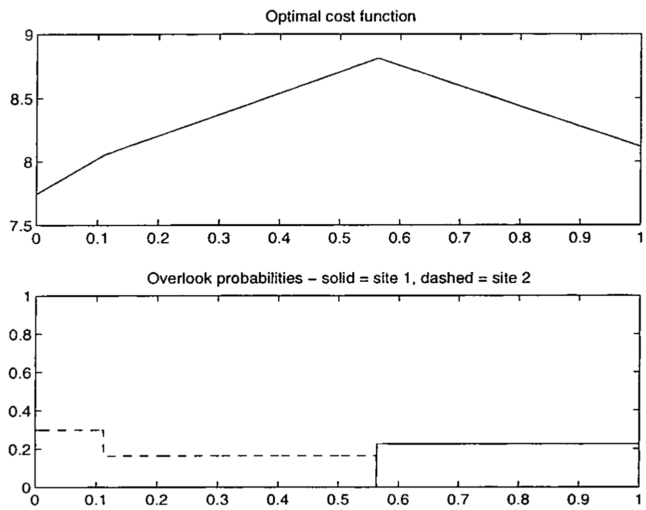


Figure 5.10: Optimal cost function with 2 levels of search intensity

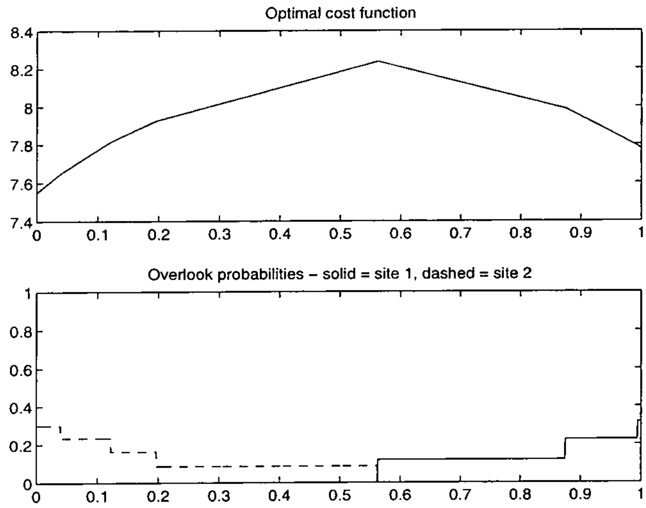


Figure 5.11: Optimal cost function with 4 levels of search intensity

This example clearly shows how the optimal policy changes as more levels of search intensity are introduced, and moreover how this affects the optimal cost function. One thing which is worth noting is that the optimal cost function tends not to change for p near 0 or p near 1. This makes sense as we are likely to be putting full search intensity into searching a given site when we believe strongly that the target is there. This leads to the natural conclusion that more use could be made of the results in Chapter 4 to calculate these functions. However, as yet it seems unclear how best to do this.

It seems like this problem is the easiest to think about in terms of development of general theory, and indeed in terms of practical application. However, as yet little of any consequence can be said about such a problem.

5.4 Machine Replacement Problems

The final problem we examine in this chapter is a Markov decision problem raised in Ross [13] concerning machine replacement. We conjecture that results already shown in Chapter 4 can be extended to this problem. It seems likely that this is only one of a whole class of problems which could be similarly examined, and possibly solved by such methods, as numerous other problems have similar structure.

5.4.1 A Simple Problem

Consider the following problem.

> A machine can be in any one of two states, *good* or *bad*. The machine produces items, which are either defective or non-defective, at the beginning of each day. The probability of a defective item is α_1 when in the good state and α_2 when in the bad state, where $\alpha_2 > \alpha_1$. Once in the bad state, the machine remains in this state until it is replaced. However, if the machine is in the good state at the beginning of one day, then with probability γ it will be in the bad state at the beginning of the next day. Each day it must be decided whether to inspect the machine at cost I or to wait. If we inspect and the machine is defective then we replace it. Let R be the cost of replacing the machine and let C be the cost incurred whenever a defective item is produced.

We say the process is in state p at time t if p is the posterior probability at time t that the machine in use is in the bad state. We further discount future costs by a factor α and the above is a Markov decision process with uncountable state space [0,1], and action space $\{inspect,\ wait\}$. From the above description, events happen in the order $\{item,$ $action,\ transition\}$. When the state is p and the action chosen is $\{wait\}$, the expected cost is:

$$(1-p)\alpha_1 C + p\alpha_2 C$$

and the next state is:

$$(1-p)\gamma + p$$

When the action chosen is $\{inspect\}$, the expected cost will be:

$$I + (1-p)\alpha_1 C + p\alpha_2(C+R)$$

and the next state will be γ as, after inspection, the machine is in the good state and γ is the probability that it turns bad overnight.

Hence the α-optimal cost function V_α will satisfy:

$$\begin{aligned} V_\alpha(p) &= \min\{(1-p)\alpha_1 C + p\alpha_2 C + \alpha V_\alpha(p + (1-p)\gamma); \\ &\qquad I + (1-p)\alpha_1 C + p\alpha_2(C+R) + \alpha V_\alpha(\gamma)\} \\ &= C(p(\alpha_2 - \alpha_1) + \alpha_1) + \min\{\alpha V_\alpha(\gamma + p(1-\gamma)); \\ &\qquad I + pR + \alpha V_\alpha(\gamma)\} \end{aligned} \tag{5.6}$$

It is known, Ross [13], that the α-optimal cost function is increasing and concave. We now show this, together with the additional fact that if the optimal policy is to $\{inspect\}$ whenever the state p is greater than or equal to some threshold P^*, then the cost function is piecewise linear.

From equation (5.6) above, the second term in the minimisation is linear in p and larger than the first term when $p = 0$. A threshold policy is optimal if our cost function V_α is increasing in p and concave and the second term is smaller than the first when $p = 1$. To show that V_α is increasing we use to the method of successive approximation: take some nondecreasing function $V_\alpha^0(p)$, substitute it in the right hand side of the optimal cost equation (5.6) and define $V_\alpha^1 = \mathcal{L}V_\alpha^0$, with $\mathcal{L}$ as in Theorem 2.2. As each of the terms in the minimisation is nondecreasing and $C(p(\alpha_2 - \alpha_1) + \alpha_1)$ is increasing it follows that $V_\alpha^1(p)$ is increasing.

Let $V_\alpha^n = \mathcal{L}V_\alpha^{n-1}$ for $n = 2, 3, \ldots$. By induction each V_α^n is an increasing function in p. Now by Theorem 2.4, our optimal cost function V_α is increasing, as we can successively approximate it, starting from the nondecreasing function

$$V_0(p) = 0\,,\ \forall p \in [0,1]\ .$$

That V_α is concave is shown by writing it as the infimum of a family of linear functions, just as in equation (3.5) and the sentence following it.

What is perhaps not known is that the optimal policy is a threshold policy (personal communication from Dr. I. MacPhee) and hence the optimal cost function is piecewise linear. This can be demonstrated if we define the threshold by

$$P^* = \min\{p : \alpha V_\alpha(\gamma + p(1-\gamma)) \geq I + pR + \alpha V_\alpha(\gamma)\} \tag{5.7}$$

and also

$$T(p) = \gamma + p(1-\gamma)$$

so $T(p)$ is linear in p, going from γ at $p = 0$ up to 1 at $p = 1$. Note also that $T(p) > p$ when $\gamma > 0$. (This is not surprising, because if $\gamma = 0$, the machine never becomes bad!)

If $P^* > 1$, the policy $\{wait\}$ is optimal for all p and V_α is linear. We now claim that when $P^* < 1$ the optimal cost function $V_\alpha(p)$ is piecewise linear, with corners at P^*, $T^{-1}(P^*)$, $T^{-2}(P^*)$, ... ,$T^{-(n-1)}(P^*)$, $T^{-n}(P^*)$, where n is such that $T^{-n}(P^*) < \gamma$ and $T^{-(n-1)}(P^*) > \gamma$. Note that n exists whenever $\gamma > 0$. Taking some point $\lambda = \gamma + \delta > \gamma$ ($\delta > 0$), $T^{-1}(\lambda) = \frac{\delta}{1-\gamma} \to 0$ as $\delta \to 0$. Hence, we can take a sufficiently small δ to ensure that $T^{-1}(\lambda) < \gamma$. Furthermore, $T^{-1}(p) < p$ when $\gamma > 0$, so starting from any point Q, $T^{-N}(Q) < \gamma$ for some N. Hence, our optimal cost function is of the form:

$$V_\alpha(p) = \begin{cases} a_{n+1}p + b_{n+1} & \text{if } p \leq T^{-n}(P^*) \\ a_n p + b_n & \text{if } T^{-n}(P^*) \leq p \leq T^{-(n-1)}(P^*) \\ \vdots \quad \vdots & \vdots \\ a_2 p + b_2 & \text{if } T^{-2}(P^*) \leq p \leq T^{-1}(P^*) \\ a_1 p + b_1 & \text{if } T^{-1}(P^*) \leq p \leq P^* \\ a_0 p + b_0 & \text{if } P^* \leq p \end{cases} \tag{5.8}$$

where the a_i and b_i are constant scalars, and the linear function $a_0 p + b_0$ is the same as the linear replacement cost in our optimal cost function equation (5.6). i.e. $V_\alpha(p) = a_0 p + b_0 = I + \alpha_1 C + p((\alpha_2 - \alpha_1)C + R) + \alpha V_\alpha(\gamma)$. Graphically, this result can be seen in figure 5.12.

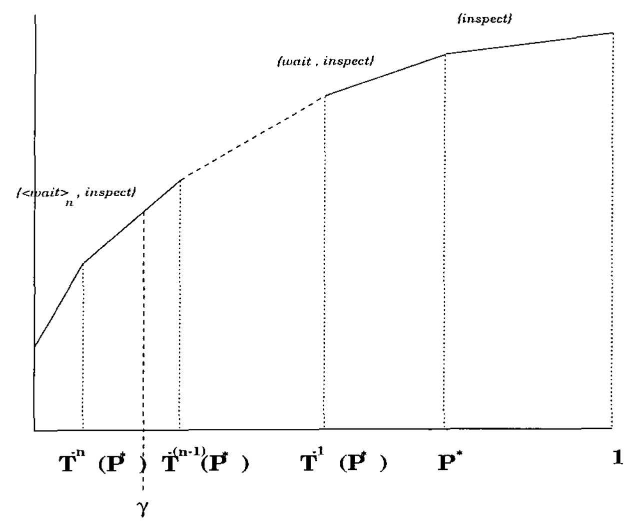


Figure 5.12: Piecewise linear optimal cost function

5.4.2 Extension of Problem

One way to extend the model is to build in Bayesian updating of our state variable and base the replacement decision on the process state. At each stage we examine the item produced and given that the machine is in state p, update our belief to $D(p)$, if the item produced was defective, and $G(p)$ if the item was non-defective, where $G(.)$ and $D(.)$ are

$$D(p) = P(\text{bad state} \mid \text{defective item})$$

$$G(p) = P(\text{bad state} \mid \text{non} - \text{defective item}) \ .$$

Using Bayes theorem we find

$$D(p) = \frac{\alpha_2 p + \gamma\alpha_1(1-p)}{\alpha_2 p + \alpha_1(1-p)}$$

and

$$G(p) = \frac{(1-\alpha_2)p + \gamma(1-\alpha_1)(1-p)}{(1-\alpha_2)p + (1-\alpha_1)(1-p)} \ .$$

It is worth noting that these, like $T(p)$ above, are both increasing in p, starting at γ when $p = 0$, and rising to 1 when $p = 1$. Moreover, applying Lemma 4.1, we find

Corollary 5.1

$G(p)$ is always increasing and convex and $D(p)$ is always increasing and concave, with $D(p) \geq G(p) \ \forall p \in [0, 1]$.

Proof

Straightforward application of Lemma 4.1, remembering that $\alpha_2 > \alpha_1$.

∎

Pictorially, we have the graphs shown in Figure 5.13 below.

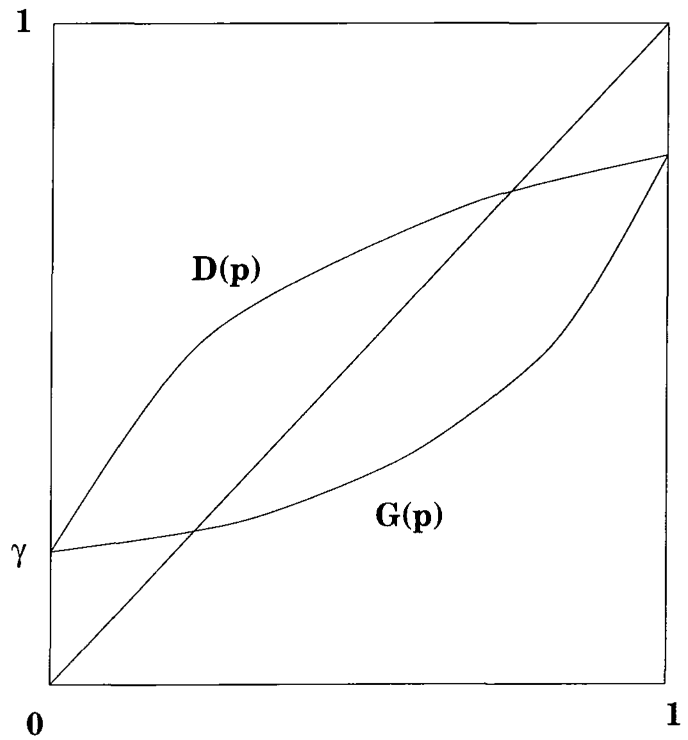


Figure 5.13: Updated beliefs, having seen item

Now we calculate the optimality equation. If we are in state p, and action $\{wait\}$ is chosen, the expected cost will be

$$p\alpha_2 C + (1-p)\alpha_1 C$$

and our next state will be $G(p)$ if we see a good item or $D(p)$ if we see a bad item where the probability of seeing a good item is

$$p(1-\alpha_2) + (1-p)(1-\alpha_1)$$

and the probability of seeing a bad item is

$$p\alpha_2 + (1-p)\alpha_1$$

If we are in state p, and we choose $\{replace\}$, the expected cost will be

$$R + C((1-\gamma)\alpha_1 + \gamma\alpha_2)$$

and our next state will be γ, as before. If we further define

$$K(p) = \alpha_1 + (\alpha_2 - \alpha_1)p$$

our optimality equation is

$$\begin{aligned} V_\alpha(p) &= \min\{C((1-p)\alpha_1 + p\alpha_2) + \alpha[(p(1-\alpha_2) + (1-p)(1-\alpha_1))V_\alpha(G(p)) \\ &\qquad + ((1-p)\alpha_1 + p\alpha_2)V_\alpha(D(p))] \; ; \\ &\quad R + C((1-\gamma)\alpha_1 + \gamma\alpha_2) + \alpha V_\alpha(\gamma)\} \\ &= \min\{CK(p) + \alpha[(1-K(p))V_\alpha(G(p)) + K(p)V_\alpha(D(p))] \; ; \\ &\quad R + CK(\gamma) + \alpha V_\alpha(\gamma)\} \end{aligned}$$

This optimality equation appears similar to that of (4.4) and it is known that the optimal policy is of threshold form. As the second term in the minimisation above is constant it suffices to show that the first term is increasing and this can be shown as in the previous subsection. Let V_α^n be the n^{th} approximation and suppose it is increasing. Now

$$V_\alpha^{n+1}(p) = \min\{M(p) \; ; \; R + CK(\gamma) + \alpha V_\alpha(\gamma)\}$$

where $M(p) = CK(p) + \alpha[(1-K(p))V_\alpha^n(G(p)) + K(p)V_\alpha^n(D(p))]$. As the function $1 - K(p)$ is differentiable we know from the intermediate value theorem that for any fixed p and $\bar{p} > p$ there exists $\hat{p} \in [p, \bar{p}]$ such that

$$(1-K(\bar{p}))V_\alpha^n(G(\bar{p})) - (1-K(p))V_\alpha^n(G(p)) = (1-K'(\hat{p}))(V_\alpha^n(G(\bar{p})) - V_\alpha^n(G(p)))$$

and from this and the facts that K, G, D and V_α^n are all increasing and $1 - K'(\hat{p}) = 1 - \alpha_2 + \alpha_1 \in (0,1)$ it follows that $M(\bar{p}) > M(p)$ i.e. M is increasing and hence V_α^{n+1} is increasing. Hence, by induction and Theorem 2.4, V_α is increasing and applying the above argument again we see that the optimal policy is indeed of theshold form.

While the motivation for piecewise linearity extends to this case, it is not easy to prove anything else. Notably, it is not possible to calculate P^* at all, as we are stuck in a *Catch-22* situation, in that in order to calculate P^*, we need to know $V_\alpha(\gamma)$, and to know $V_\alpha(\gamma)$ we need to know P^*! It would be nice to think that we could apply methods similar to those used in Chapter 4 to this problem. However, the problem we encounter is similar to that discussed in the previous two sections, in that we have an infinite number of possible scenarios into which we could fall. While we can solve the problem exactly for simple cases, such as ones where we never replace or we replace as soon as we see a bad item, it is not clear what happens as soon as we have more than 2 linear pieces to the optimal cost function as we can no longer be sure which section the value of γ will lie on. It is for this reason that an approach like the one in Chapter 4 is not feasible. Moreover, even if we could work out all the possible scenarios which could exist, we would need to completely solve each given problem in order to determine which one we lay in. However, we can conjecture the following.

Conjecture 5.5

The optimal cost function in a machine replacement problem as described above is piecewise linear. ∎

This conjecture is again borne out by numerical experiment using the Matlab program **replacement.m** as found in Appendix A. This allows us to solve approximately the optimality equation 5.6 above.

Example 5.3

If we take the following parameter values

$$\gamma = 0.25,\ C = 3,\ R = 3,\ \alpha_1 = 0.1,\ \alpha_2 = 0.8$$

we find that $P^* \approx 0.7404$. Graphically, the optimal cost function and $G(p)$ and $D(p)$ can be seen below.

∎

Superficially, we could imagine such a problem as a 2-site search problem with overlook probabilities $1 - \alpha_1$ and $1 - \alpha_2$. Indeed, if our aim was to replace whenever we saw a bad item, it could be fitted into such a format. However, under the given framework, we cannot make this sort of a trick work. The problem arises from the fact that in the Machine Replacement problem we make decisions about our next action after receiving information (in this case after observing the latest item). In the previous search problems we specified the control of the problem at the start (i.e. we decided what our actions were going to be at all times), and we receive no information during the searching process. As a result we encounter a further problem in that individual policies make little sense, as there are 2 results we can get at any stage in that we can either see a good item or a bad item. However, $D(p)$ and $G(p)$ can work in opposite directions (i.e. $D(p) > p > G(p)$ for some p), meaning that at certain stages, we could have the policy $\{$*replace if we see a bad item, but if we see a good item, then only if we see two bad items in a row*$\}$. It seems likely from the formulae that there will always be corners at $D^{-1}(P^*)$, $D^{-2}(P^*)$ etc., but other corners may well exist. It is for reasons like this that this problem remains open and unsolved.

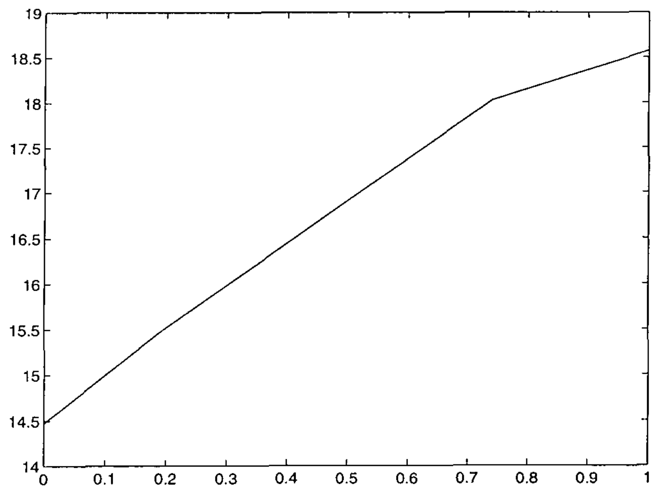


Figure 5.14: Optimal cost function $V_\alpha(p)$ with parameter values as in Example 5.4

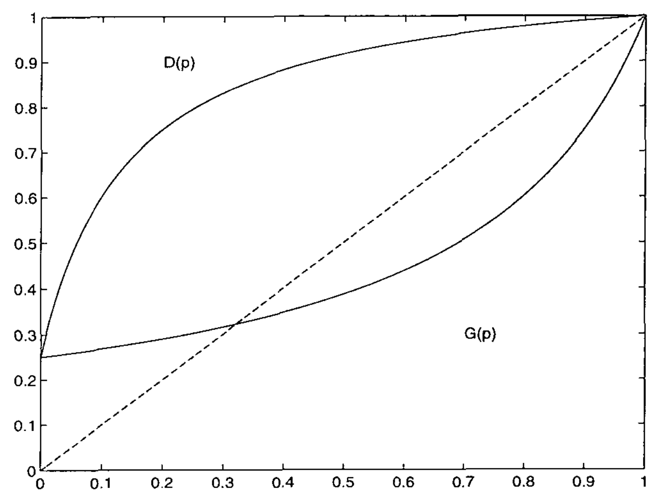


Figure 5.15: $G(p)$ and $D(p)$ for parameter values as in Example 5.4

5.5 Conclusion

In this chapter we have examined some still-unsolved problems and given evidence as to what their solution might be without being able to complete a full proof. This work might be regarded as a starting point for further research and should prove useful in the long term research of Search theory and other Markov decision process based problems. Certainly it seems possible that more could be made of the solution to the varying resource problem by Assaf and Sharlin-Bilitzky [2] which has not been attempted, notably and attempt to verify some of the possible optimal policies outlined in that paper. Such problems represent a better attempt to successfully model real problems and will hopefully lead to new results.

Chapter 6

Conclusion

The way you walk was thorny through no fault of your own, but as the rain enters the soil and the river enters the sea so tears run to a predestined end.

Maleva (Maria Ouspenskaya) - The Wolf Man (1941)

In this thesis, we have examined a variety of different search problems and given solutions to some and conjectured possible solutions for others. One thing, however, which has characterised all of this work has been the simplicity of the problems themselves – we have worked almost uniquely in two sites, and moreover the target motion is the simplest possible kind. However, I feel the research bears testimony to the fact that such problems are not as simple as they might at first appear. Over the course of doing this research, I have attended numerous conferences and met many people, and I have often been asked to explain what it is that I do. In reply, I have generally quoted the motivational example given in the preface, together with a brief explanation of the exact nature of the solution. During one interview with a Bonds Salesman from BZW I was asked this question to which I gave my usual reply. The interviewer seemed somewhat taken aback by this answer and asked in an incredulous voice 'and it's taken you *three years* to do this?'. Essentially, this highlights a major point, which is that although the problems are simple in nature, their solution is complicated and the proof thereof is even more complex [c.f. quotation, Chapter 1]. This point has always been the major stumbling block in the development of search theory and is reflected in the lack of theoretic literature on the subject. Over the last three years I feel a lot has been achieved and I hope that new theory is developed to add onto this work.

What then, of the future of search theory? To my mind, the problems studied do have real applications and these should be developed and exploited as far as possible. While I admit to not being an expert on the practical application of search theory I cannot help but feel that more could be done to bridge the divide between search theory and the practical task of searching. This disparity was brought home to me during a series of seminars I attended in 1994, the first of which was given by a member of the coast guard, and the second by a mountain rescue search team leader. The naivety of some of the arguments used for policy decisions was backed up by the harsh reality that the chances of being found by a search-and-rescue team are next to nothing. My own feeling is that, as I have discovered, the theory is very hard to develop, and moreover that in a realistic scenario, trying to check which of the possible thousands of cases we lie in would be impractical and very time consuming. As in all real life problems, we have to balance out the desire for exact results against the time taken to achieve them. It is for this reason that full value iteration is not viable for large scale problems.

The vision I have is for a future in which the (relatively) powerful computer facilities available to searchers are used in an efficient fashion to help them in their task. This, of course, involves a shift from closed-form analytic solutions to algorithmic work. Development have already been made in this direction, as work on the FAB algorithm by Washburn [17] and more recent work by Thomas and Eagle [16] on approximate solutions using path constrained algorithms show. However, much remains to be done. Most importantly, one must always remember that at the end of the day it is not mathematicians or computer scientists who will be performing the searches. As much effort has to be put into the development of a good user interface as to the algorithms themselves, for a computer is only as good as its operator lets it be.

In conclusion, search theory is a wide ranging and multi-functional branch of mathematics. It has been studied in many ways and a great deal has been achieved. At present it stands on the threshold of a new approach. I hope that the work of this thesis helps to achieve new and better results.

Appendix A

Matlab Programs

A.1 diffalpha.m

This program takes in parameter values, and calculates the optimal cost function for a 2-site search problem via. iteration. It outputs three graphs

(i) A graph of the optimal cost function $V(p)$ against p.

(ii) A graph of $L_1(p)$ and $L_2(p)$ against p.

(iii) A graph comparing the cost of $1, \langle d \rangle$ with $2, \langle d \rangle$, where d represents the policy suggested by the optimal cost function $V(p)$.

We can use the values of p1, p2 and q1 as output by the program to tell which case we are in. These values correspond to the values of P_1, P_2 and P^* as described in Chapter 4. If we find we are in Case 4, we can use the values of L12 and L21 to determine which subcase we are in.

```
clear
hold off
n=input('how many points? ');
a=input('P(box 1|box 1) ? ');
```

```
b=input('P(box 2|box 2) ? ');
alpha1=input('overlook probability for house 1 ');
alpha2=input('overlook probability for house 2 ');
c1=input('cost per look for house 1? ');
c2=input('cost per look for house 2? ');
v0=zeros(n+1,1);
v1=v0;
v0(1)=1;

i=1:n+1;
basis=(i-1)/n;

t1=round(n*(a*alpha1*basis + (1-b)*(1-basis))./(alpha1*basis + 1-basis));
t2=round(n*(alpha2*(1-b)*(1-basis) + a*basis)./(alpha2*(1-basis)+basis));

while max(abs(v1-v0))>0.00001;
        v0=v1;
        v1=min(c1+ v0(t1)'.*(alpha1*basis+ (1-basis)) ,
                c2 +v0(t2)'.*(alpha2*(1-basis) +basis))';
end

plot(basis,v1)
hold on
q1=max(basis'.*(v1==max(v1)))
pause

C=[0,1,0, 1];
axis(C)
        tt1=t1/n;
        tt2=t2/n;
        ttt1=v1(t1).*(alpha1*basis+1-basis)';
        ttt2=v1(t2).*(alpha2*(1-basis)+basis)';
L1=basis(round(n*(a*alpha1*q1 + (1-b)*(1-q1))./(alpha1*q1 + 1-q1)));
L2=basis(round(n*(alpha2*(1-b)*(1-q1) + a*q1)./(alpha2*(1-q1)+q1)));
L21=basis(round(n*(a*alpha1*L2 + (1-b)*(1-L2))./(alpha1*L2 + 1-L2)));
L12=basis(round(n*(alpha2*(1-b)*(1-L1) + a*L1)./(alpha2*(1-L1)+L1)));

plot(basis,tt1)
hold on
plot(basis,tt2,'g')
```

```
plot(basis,basis,'--w')
pause

hold off

plot(basis,ttt1)
hold on
plot(basis,ttt2,'g')
pp1=(2-alpha1*a-b-sqrt((alpha1*a+b)^2-4*alpha1*(a+b-1)))
        /(2*(1-alpha1))
pp2=(a-alpha2*(2-b)+sqrt((a-alpha2*(2-b))^2 +4*alpha2*(1-alpha2)*(1-b)))
        /(2*(1-alpha2))
Delta=a+b-1;
L12
L21
q1
```

A.2 leprechaun2.m

This program calculates the optimal cost function and first look optimal policy regions for the 3-site problem as discussed in Section 5.1. It creates a lattice of points, and then applies value iteration to approximate the optimal cost function. For each updated point it calculates the triangle of lattice nodes which contains that point and then expresses the original point as the weighted average of those three nodes. We encounter certain problems with this approach, notably the decision as to which triangle a given point lies in. This is highlighted by the diagram below.

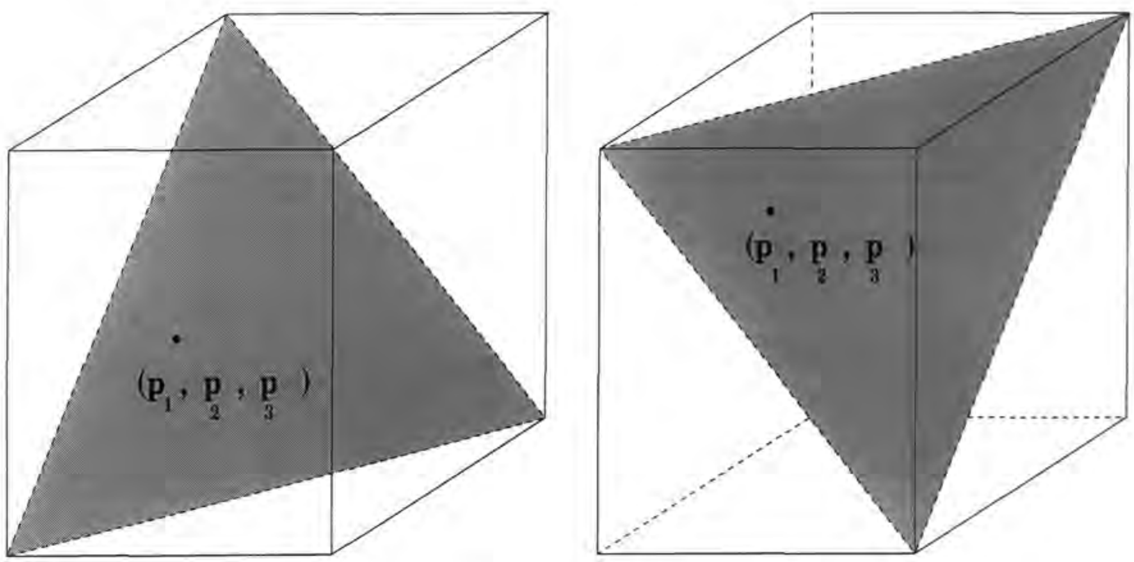


Figure A.1: Possible triangles which can exist within the cube from *floor*(p) to *ceiling*(p)

The further problem that this triangulation raises is that it has a tendency to 'smooth' out ridges and corners. This brings questions about the accuracy of this function to the true optimal cost function, which remain unclear. What we can say about the numerical result is that it will be concave, as the effect of the triangulation produces a function which always lies below the true optimal cost function.

```
clear
n=input('how many points per side?');

% First we create the basis matrix in a logical fashion.

a=[0:n];
b=ones(size(a));
AB=1/n*a'*b;
L=tril(ones(n+1));
G1=triu(AB)';
G2=rot90(G1,3);
M(:,1)=G1(L);
M(:,2)=G2(rot90(L,3));
M(:,3)=1-M(:,1)-M(:,2);
clear AB;clear a;clear b;clear G1;clear G2;clear L;
pack

% Put in the basic variables - transition matrix & overlook prob.

priors

% calculating new prior probabilities of where
% leprechaun is using Bayes' theorem

M01=M; M02=M; M03=M; M01(:,1)=z*M(:,1); M02(:,2)=z*M(:,2); M03(:,3)=z*M(:,3);

% working out prob(look in house i & don't see leprechaun)=hi

h1=(sum(M01'))'; h2=(sum(M02'))'; h3=(sum(M03'))';

M11=M01*A./[h1 h1 h1]; M12=M02*A./[h2 h2 h2]; M13=M03*A./[h3 h3 h3];
%M11(0.5*n*(n+3) + 1,:)=[1 0 0];
%M12(n+1,:)=[0 1 0];
%M13(1,:)=[0 0 1];

clear M01; clear M02; clear M03;
%clear h1; clear h2; clear h3;
pack

% Now we work out which triangle in our triangulation contains the updated
```

```
%point, using a boolean variable.

R1=[1/n*ceil(n*M11(:,1)), 1/n*(floor(n*M11(:,2)))];
R2=[1/n*ceil(n*M12(:,1)), 1/n*(floor(n*M12(:,2)))];
R3=[1/n*ceil(n*M13(:,1)), 1/n*(floor(n*M13(:,2)))];

test11=0.5*(n*R1(:,1)).*(2*n + 3 - n*R1(:,1)) + n*R1(:,2) +1;
test21=0.5*(n*R2(:,1)).*(2*n + 3 - n*R2(:,1)) + n*R2(:,2) +1;
test31=0.5*(n*R3(:,1)).*(2*n + 3 - n*R3(:,1)) + n*R3(:,2) +1;
test12=round(0.5*(n*R1(:,1) -1).*(R1(:,1)>0).*(2*n + 3 - n*R1(:,1) +1) + n*R1(:,2) + 1);
test22=round(0.5*(n*R2(:,1) -1).*(R2(:,1)>0).*(2*n + 3 - n*R2(:,1) +1) + n*R2(:,2) + 1);
test32=round(0.5*(n*R3(:,1) -1).*(R3(:,1)>0).*(2*n + 3 - n*R3(:,1) +1) + n*R3(:,2) + 1);
test14=round(0.5*(n*R1(:,1) -1).*(R1(:,1)>0).*(2*n + 3 - n*R1(:,1) +1) + n*R1(:,2)) + 2;
test24=round(0.5*(n*R2(:,1) -1).*(R2(:,1)>0).*(2*n + 3 - n*R2(:,1) +1) + n*R2(:,2)) + 2;
test34=round(0.5*(n*R3(:,1) -1).*(R3(:,1)>0).*(R1(:,1)>0).*(2*n + 3 - n*R3(:,1) +1) + n*R3(:,2)) + 2;
test15=0.5*(n*R1(:,1)).*(2*n + 3 - n*R1(:,1)) + n*R1(:,2) +2;
test25=0.5*(n*R2(:,1)).*(2*n + 3 - n*R2(:,1)) + n*R2(:,2) +2;
test35=0.5*(n*R3(:,1)).*(2*n + 3 - n*R3(:,1)) + n*R3(:,2) +2;
l1=[test11 , test12 , test14 ];
l2=[test21 , test22 , test24 ];
l3=[test31 , test32 , test34 ];
ll1=[test15 , test11 , test14];
ll2=[test25 , test21 , test24];
ll3=[test35 , test31 , test34];
lll1=[test11, test14, test11+0.5*n*(n+1)];
lll2=[test21, test24, test21+0.5*n*(n+1)] ;
lll3=[test31, test34, test31+0.5*n*(n+1)];
test16=(abs((1-R1(:,1)-R1(:,2))-(1/n*floor(n*(1-M11(:,1)-M11(:,2)))))<=0.0001);
test26=(abs((1-R2(:,1)-R2(:,2))-(1/n*floor(n*(1-M12(:,1)-M12(:,2)))))<=0.0001);
test36=(abs((1-R3(:,1)-R3(:,2))-(1/n*floor(n*(1-M13(:,1)-M13(:,2)))))<=0.0001);
for i=1:3
Tri1(:,i)=l1(:,i).*test16.*(R1(:,1)>0.00001) + ll1(:,i).*(1-test16).*(R1(:,1)>0.00001)
                + lll1(:,i).*(R1(:,1)<=0.00001);
Tri2(:,i)=l2(:,i).*test26.*(R2(:,1)>0.00001) + ll2(:,i).*(1-test26).*(R2(:,1)>0.00001)
                + lll2(:,i).*(R2(:,1)<=0.00001);
Tri3(:,i)=l3(:,i).*test36.*(R3(:,1)>0.00001) + ll3(:,i).*(1-test36).*(R3(:,1)>0.00001)
                + lll3(:,i).*(R3(:,1)<=0.00001);
end
end
end
```

```
clear R1; clear R2; clear R3; clear test11; clear test21; clear test31;
clear test12; clear test22; clear test32; clear test14; clear test24; clear test34;
clear test15; clear test25; clear test35; clear test16; clear test26; clear test36;
clear l1; clear l2; clear l3; clear l11; clear l12; clear l13;
clear l111; clear l112; clear l113;
pack

l=1:1.5*(n+1)*(n+2);
ll=1:2*(n+1)*(n+2);
i1=[l;l;l;l];
i=reshape(i1,6*(n+1)*(n+2),1);
i2=reshape(ll,4,0.5*(n+1)*(n+2));
i3=[i2' i2' i2'];
j=reshape(i3',6*(n+1)*(n+2),1);


k1=reshape([M(Tri1(:,1),:) ones(0.5*(n+1)*(n+2),1)  M(Tri1(:,2),:)  ones(0.5*(n+1)*(n+2),1)  M(Tri1(:,3),:)
        ones(0.5*(n+1)*(n+2),1)]',6*(n+1)*(n+2),1);
S1=sparse(j,i,k1);
k2=reshape([M(Tri2(:,1),:) ones(0.5*(n+1)*(n+2),1)  M(Tri2(:,2),:)  ones(0.5*(n+1)*(n+2),1)  M(Tri2(:,3),:)
        ones(0.5*(n+1)*(n+2),1)]',6*(n+1)*(n+2),1);
S2=sparse(j,i,k2);
k3=reshape([M(Tri3(:,1),:) ones(0.5*(n+1)*(n+2),1)  M(Tri3(:,2),:)  ones(0.5*(n+1)*(n+2),1)  M(Tri3(:,3),:)
         ones(0.5*(n+1)*(n+2),1)]',6*(n+1)*(n+2),1);
S3=sparse(j,i,k3);

M11(:,4)=ones(size(M11(:,3)));
M12(:,4)=M11(:,4);
M13(:,4)=M11(:,4);
D1=reshape(M11',2*(n+1)*(n+2),1);
D2=reshape(M12',2*(n+1)*(n+2),1);
D3=reshape(M13',2*(n+1)*(n+2),1);


W1=(reshape(S1\D1,3,0.5*(n+1)*(n+2)))';
W2=(reshape(S2\D2,3,0.5*(n+1)*(n+2)))';
W3=(reshape(S3\D3,3,0.5*(n+1)*(n+2)))';

clear D1; clear D2; clear D3;
%clear S1; clear S2; clear S3;
```

```
clear k1; clear k2; clear k3; clear i1; clear i2; clear i3; clear i; clear j; clear l; clear l1;
pack

% Now we can iterate on the optimality equation!

v0=ones(size(h1));
v1=zeros(size(v0));
counter=0;
while max(abs(v1-v0))>=0.00001
v0=v1;
counter=counter+1;
trial1=(v0(Tri1(:,1)).*W1(:,1) + v0(Tri1(:,2)).*W1(:,2) + v0(Tri1(:,3)).*W1(:,3)).*h1;
trial2=(v0(Tri2(:,1)).*W2(:,1) + v0(Tri2(:,2)).*W2(:,2) + v0(Tri2(:,3)).*W2(:,3)).*h2;
trial3=(v0(Tri3(:,1)).*W3(:,1) + v0(Tri3(:,2)).*W3(:,2) + v0(Tri3(:,3)).*W3(:,3)).*h3;

v1=c+(min([trial1 trial2 trial3]'))';
end
eps=0.00001;
trial4= (abs(trial1-(v1-1))<eps) + 2*(abs(trial2-(v1-1))<eps) + 3*(abs(trial3-(v1-1))<eps);
% - 3*(trial1==trial2).*((v1-1)==trial1) - 4*(trial1==trial3).*((v1-1)==trial1)
%- 5*(trial2==trial3).*(trial2==(v1-1)) + 6*(trial1==trial2).*(trial2==trial3);


test=ones(size(v1))*max(v1);
test2=[1:0.5*(n+1)*(n+2)]';
test3=test2.*(test==v1);
M(max(test3),:)


% Now we can draw a mesh plot of the optimal cost.

clg
hold off
meshdom(1:n+1, 1:n+1);

ss=round(n*M(:,1)+1);
tt=round(n*M(:,2) +1);
vv1=v1-min(v1);
ZZ=sparse(ss,tt,vv1);
Z=full(ZZ);
```

```
end
end
%for N=30:60:390;
mesh (Z,[30,30])
axis([0,n+1,0,n+1,0,max(v1)-min(v1)])
pause(0.5)
end
title('meshplot of expected searchtimes')
pause

%lets look at the optimal first-look strategies

%test1=M(trial4>0,:);
test2=M.*[trial4==1 trial4==1 trial4==1];
test3=M.*[trial4==2 trial4==2 trial4==2];
test4=M.*[trial4==3 trial4==3 trial4==3];
clg
plot(test2(:,1), test2(:,2),'*r')
hold on
plot(test3(:,1), test3(:,2),'xy')
plot(test4(:,1), test4(:,2),'ob')
title('*=house1,x=house2,o=house3')
```

A.3 resources.m

This program approximates the optimal cost function for the varying resource search problem as found in section 5.2. I takes in parameter values, together with the number of possible divisions of search effort and outputs 2 graphs, one above the other. These represent

(i) A graph of the approximate optimal cost function

(ii) A graph of the overlook probabilities for searches of sites 1 and 2

It can operate with any number of search levels above 0.

```
% In this program ,we have some fixed amount of resources, and we can divide
% them between house searches, with b1 of the resource going to 1 , and
% b2 to house2.
% First we input the basic variables
clear
clg
hold off
n=1000;
a=input('P(box 1|box 1) ? ');
b=input('P(box 2|box 2) ? ');
c1=input('Cost of full search for site 1 ? ');
c2=input('Cost of full search for site 2 ? ');
c3=input('Cost of doing nothing ? ');
Z1=input('Basic overlook for site 1 ? ');
Z2=input('Basic overlook for site 2 ? ');
m=input('Number of levels of search (amount of resource available)? ');



%z1=1-Z1;
%z2=1-Z2;

ss=[0:m];
tt=ones(size(ss));
AB=1/m*ss'*tt;
```

```
L=tril(ones(m+1));
G1=triu(AB)';
G2=rot90(G1,3);
s=G1(L);
t=G2(rot90(L,3));

C=((1-s-t)*c3+s*c2+t*c1);
%alpha2=z2*t;
%alpha1=z1*s;
for i=1:(m+1)*(m+2)/2
alpha1(i)=Z1^s(i);
alpha2(i)=Z2^t(i);
end
alpha1=(1-alpha1)';
alpha2=(1-alpha2)';



v0=zeros(n+1,1);
v1=v0;
v0(1)=1;

i=1:n+1;
basis=(i-1)/n;

for j=2:0.5*(m+1)*(m+2)
tt1=(a*(1-alpha1(j))*basis + (1-b)*(1-alpha2(j))*(1-basis));
tt2=((1-alpha1(j))*basis + (1-alpha2(j))*(1-basis))';
u(j,:)=(tt1./(tt2'));
        if alpha1(j)==1;
        u(j,n+1)=a;
        end
                if alpha2(j)==1;
                u(j,1)=1-b;
                end
end
u(1,:)=basis;
u=1+round(n*u);

for i=1:0.5*(m+1)*(m+2)
```

```
plot(basis,u(i,:))
hold on
end

while max(abs(v1-v0))>0.00001;
        v0=v1;
        for i=1:0.5*(m+1)*(m+2)
        v(i,:)=C(i)+v0(u(i,:))'.*((1-alpha1(i))*basis + (1-alpha2(i))*(1-basis));
        end
        [vv,l]=min(v);
        v1=vv';
end
hold off
while max(abs(v1-v0))>0.00001;
        v0=v1;
        for i=1:0.5*(m+1)*(m+2)
        v(i,:)=C(i)+v0(u(i,:))'.*((1-alpha1(i))*basis + (1-alpha2(i))*(1-basis));
        end
        [vv,l]=min(v);
        v1=vv';
end
hold off

subplot(2,1,1),plot(basis,v1);
title('Optimal cost function')

subplot(2,1,2),plot(basis,alpha1(l));
hold on
plot(basis,alpha2(l),'--r');
axis([0 1 0 1]);
title('Overlook probabilities - solid = site 1, dashed = site 2')
for c=2:n+1
test(c)=l(c)*(abs(l(c)-l(c-1))>0);
end
test(1)=l(1);
sparse(test)
```

A.4 replacement.m

This program approximately solves the machine replacement problem discussed in Chapter 5. It takes in parameter values and outputs two graphs

i A graph of the optimal cost function $V_\alpha(p)$

ii A graph of $G(p)$ and $D(p)$

It also outputs the approximate value of P^*.

```
clear
hold off
n=10000;
gamma=input('value of gamma (probability of good to bad) ? ');
c=input('cost per look ? ');
r=input('cost of defective item ? ');
alpha=input('discount factor ? ');
b=input('probability of bad item in bad state ? ');
a=input('probability of bad item in good state ? ');
v0=zeros(n+1,1);
v1=v0;
t1=v0;
t2=t1;
v0(1)=1;
i=1:n+1;
        basis=(i-1)/n;
        basiss=basis;

        d=1+round(n*(b*basis + gamma*a*(1-basis))./(b*basis + a*(1-basis)));
        g=1+round(n*((1-b)*basis + gamma*(1-a)*(1-basis))./((1-b)*basis + (1-a)*(1-basis)));
if b==1;
d(:,1)=1+round(n*gamma);
end
if a==0;
g(:,n+1)=n+1;
end
        k=(basis*b + (1-basis)*a);
```

```
while max(abs(v1-v0))>0.00001;
        v0=v1;
        v1=min((c*k(1+(n*basis)))' + (alpha * ((v0(g,:).*(1-((k(1+(n*basis)))')))+ v0(d,:).*((k(1+(n*basis))))')),
 r+c*(k(1+(n*basis)))' + alpha*v0(1+(n*gamma),:));
end
end
look=(c*k(1+(n*basis)))' + (alpha * ((v0(g,:).*(1-((k(1+(n*basis)))')))+ v0(d,:).*((k(1+(n*basis))))'));
repl=r+c*(k(1+(n*basis)))' + alpha*v0(1+(n*gamma),:);
limit=r+c*a + alpha*v0(1+(n*gamma),:);
plot(basis,v1)
end
end
end
end
p=max(basis.*((abs(look-repl))==min(abs(look-repl)))')
```

Appendix B

Maple Calculations

B.1 Boundary between Regions 1 and 3

First, we need to calculate the values of $P1$, $P3$ and P_2.

A1:=-C2/(1-b);

$$A1 := -\frac{C2}{1-b}$$

B1:=C1/(1-b)*(2-b-a*z1)/(1-z1);

$$B1 := \frac{C1\,(2-b-a\,z1)}{(1-b)\,(1-z1)}$$

A2:=C2/(1-a);

$$A2 := \frac{C2}{1-a}$$

B2:=C2/(1-a)*(1-a+z2*(1-b))/(1-z2);

$$B2 := \frac{C2\,(1-a+z2\,(1-b))}{(1-a)\,(1-z2)}$$

P3:=(B2-B1)/(A1-A2);

$$P3 := \frac{\dfrac{C2\,(1-a+z2\,(1-b))}{(1-a)\,(1-z2)} - \dfrac{C1\,(2-b-a\,z1)}{(1-b)\,(1-z1)}}{-\dfrac{C1}{1-b} - \dfrac{C2}{1-a}}$$

simplify(P3);

$$(-2\,C1 + C2 - C2\,z1 - C2\,b - C2\,a + C2\,z2 + 2\,C1\,z2 + 2\,C1\,a + C1\,b + C2\,b\,z1 + C2\,a\,z1 + C2\,a\,b - C2\,z2\,z1 - 2\,C2\,z2\,b + C2\,z2\,b^2 - 2\,C1\,a\,z2 - C1\,z2\,b - C1\,a\,b + C1\,a\,z1 - C1\,a^2\,z1 - C2\,a\,b\,z1 + 2\,C2\,z2\,b\,z1 - C2\,z2\,b^2\,z1 + C1\,a\,b\,z2 - C1\,a\,z1\,z2 + C1\,a^2\,z1\,z2) \Big/ ((-C1 + C1\,a - C2 + C2\,b)(-1 + z1)(-1 + z2))$$

P1:=(2-z1*a-b-z2*(1-z1*(a+b-1))-C2/C1*(1-b)*(1-z1))/(2-z1-z2-(a+b-1)*(1-z1*z2));

$$P1 := \frac{2 - a\,z1 - b - z2\,(1 - z1\,(a+b-1)) - \dfrac{C2\,(1-b)\,(1-z1)}{C1}}{2 - z1 - z2 - (a+b-1)\,(1 - z2\,z1)}$$

P2:=(a-z2*(2-b)+sqrt((a-z2*(2-b))^2+4*z2*(1-z2)*(1-b)))/(2*(1-z2));

$$P\_2 := \left(a - z2\,(2-b) + \sqrt{a^2 - 4\,a\,z2 + 2\,a\,b\,z2 + z2^2\,b^2 + 4\,z2 - 4\,z2\,b}\right) \Big/ (2 - 2\,z2)$$

Now, we check that the contours along which $P1 = P_2$ and $P3 = P_2$ are the same.

contour1:=solve(P1=P2,b);

$$contour1 := \mathrm{RootOf}(3\,C2\,C1\,a + C1^2\,z1^2\,z2^2 - 4\,C1\,C2 - 6\,C1^2\,a + 2\,C1^2\,a^2 + C2^2 - 2\,C2^2\,z1 - C2^2\,z2 + C2^2\,z1^2 + 4\,C1^2 + 7\,C1^2\,a\,z2 + 2\,C2^2\,z2\,z1 - C2^2\,z1^2\,z2 - 2\,C1^2\,a^2\,z2 + 4\,C1\,C2\,z1 - C2\,C1\,a^2 - 2\,C1^2\,a\,z1 - 2\,C1^2\,z1\,z2 + 3\,C1^2\,a^2\,z1 - C1^2\,a^3\,z1 + 6\,C1^2\,a\,z1\,z2 - 5\,C1^2\,a^2\,z1\,z2 - C2\,z1^2\,C1\,a - 4\,C2\,z1^2\,C1\,z2 + 5\,C2\,z1^2\,C1\,a\,z2 - C2\,z1^2\,C1\,a^2\,z2 - 2\,C2\,z1\,C1\,a + C2\,z1\,C1\,a^2 - 6\,C2\,z1\,C1\,a\,z2 + C1\,a^2\,C2\,z2\,z1 + 4\,C1\,C2\,z2\,z1 + C2\,C1\,a\,z2 + C1^2\,a^3\,z1\,z2 - 4\,C1^2\,a\,z2^2\,z1$$

$$
\begin{aligned}
&+ 2\, C1^2\, a^2\, z2^2\, z1 - C1^2\, a\, z2^2 + 3\, C1^2\, a\, z1^2\, z2 - 3\, C1^2\, a^2\, z1^2\, z2 \\
&+ C1^2\, a^3\, z1^2\, z2 - 3\, C1^2\, z1^2\, z2^2\, a + 3\, C1^2\, z1^2\, z2^2\, a^2 \\
&- C1^2\, a^3\, z1^2\, z2^2 + z2^2\, C1^2 - 5\, z2\, C1^2 + 2\, z2^2\, C1^2\, z1 - z2\, C1^2\, z1^2 \\
&+ (-C1\, C2\, z2\, z1 + C2\, z2^2\, z1^2\, C1 + C1\, C2\, z2 - C2\, z2^2\, C1\, z1)\,\_Z^3 \\
&+ (C2\, C1\, a - 2\, C1\, C2 - C1^2\, a + C2^2 - 2\, C2^2\, z1 - C2^2\, z2 + C2^2\, z1^2 \\
&+ C1^2 + C1^2\, a\, z2 + 2\, C2^2\, z2\, z1 - C2^2\, z1^2\, z2 + 2\, C1\, C2\, z1 \\
&- 4\, C1\, C2\, z2 - C1^2\, z1\, z2 + C1^2\, a\, z1\, z2 - 3\, C2\, z1^2\, C1\, z2 \\
&+ C2\, z1^2\, C1\, a\, z2 - C2\, z1\, C1\, a - 2\, C2\, z1\, C1\, a\, z2 - C2\, z2^2\, z1\, C1\, a \\
&+ 7\, C1\, C2\, z2\, z1 + C2\, C1\, a\, z2 + C2\, z2^2\, C1\, z1 - 2\, C2\, z2^2\, z1^2\, C1 \\
&+ C2\, z2^2\, z1^2\, C1\, a - C1^2\, a\, z2^2\, z1 - z2\, C1^2 + z2^2\, C1^2\, z1 \\
&+ C2\, z2^2\, C1)\,\_Z^2 + (-4\, C2\, C1\, a - C1^2\, z1^2\, z2^2 + 6\, C1\, C2 \\
&+ 5\, C1^2\, a - C1^2\, a^2 - 2\, C2^2 + 4\, C2^2\, z1 + 2\, C2^2\, z2 - 2\, C2^2\, z1^2 \\
&- 4\, C1^2 - 6\, C1^2\, a\, z2 - 4\, C2^2\, z2\, z1 + 2\, C2^2\, z1^2\, z2 + C1^2\, a^2\, z2 \\
&- 6\, C1\, C2\, z1 + 3\, C1\, C2\, z2 + C2\, C1\, a^2 + C1^2\, a\, z1 + 2\, C1^2\, z1\, z2 \\
&- C1^2\, a^2\, z1 - 4\, C1^2\, a\, z1\, z2 + 2\, C1^2\, a^2\, z1\, z2 + C2\, z1^2\, C1\, a \\
&+ 7\, C2\, z1^2\, C1\, z2 - 6\, C2\, z1^2\, C1\, a\, z2 + C2\, z1^2\, C1\, a^2\, z2 \\
&+ 3\, C2\, z1\, C1\, a - C2\, z1\, C1\, a^2 + 8\, C2\, z1\, C1\, a\, z2 - C1\, a^2\, C2\, z2\, z1 \\
&+ C2\, z2^2\, z1\, C1\, a - 10\, C1\, C2\, z2\, z1 - 2\, C2\, C1\, a\, z2 + C2\, z2^2\, z1^2\, C1 \\
&- C2\, z2^2\, z1^2\, C1\, a + 3\, C1^2\, a\, z2^2\, z1 - C1^2\, a^2\, z2^2\, z1 + C1^2\, a\, z2^2 \\
&- 2\, C1^2\, a\, z1^2\, z2 + C1^2\, a^2\, z1^2\, z2 + 2\, C1^2\, z1^2\, z2^2\, a - C1^2\, z1^2\, z2^2\, a^2 \\
&- z2^2\, C1^2 + 5\, z2\, C1^2 - 2\, z2^2\, C1^2\, z1 + z2\, C1^2\, z1^2 - C2\, z2^2\, C1)\,\_Z)
\end{aligned}
$$

contour2:=solve(P3=P2,b);

$$
\begin{aligned}
\mathit{contour2} := \mathrm{RootOf}(&3\,C2\,C1\,a + C1^2\,z1^2\,z2^2 - 4\,C1\,C2 - 6\,C1^2\,a \\
&+ 2\,C1^2\,a^2 + C2^2 - 2\,C2^2\,z1 - C2^2\,z2 + C2^2\,z1^2 + 4\,C1^2 \\
&+ 7\,C1^2\,a\,z2 + 2\,C2^2\,z2\,z1 - C2^2\,z1^2\,z2 - 2\,C1^2\,a^2\,z2 + 4\,C1\,C2\,z1 \\
&- C2\,C1\,a^2 - 2\,C1^2\,a\,z1 - 2\,C1^2\,z1\,z2 + 3\,C1^2\,a^2\,z1 - C1^2\,a^3\,z1 \\
&+ 6\,C1^2\,a\,z1\,z2 - 5\,C1^2\,a^2\,z1\,z2 - C2\,z1^2\,C1\,a - 4\,C2\,z1^2\,C1\,z2 \\
&+ 5\,C2\,z1^2\,C1\,a\,z2 - C2\,z1^2\,C1\,a^2\,z2 - 2\,C2\,z1\,C1\,a \\
&+ C2\,z1\,C1\,a^2 - 6\,C2\,z1\,C1\,a\,z2 + C1\,a^2\,C2\,z2\,z1 \\
&+ 4\,C1\,C2\,z2\,z1 + C2\,C1\,a\,z2 + C1^2\,a^3\,z1\,z2 - 4\,C1^2\,a\,z2^2\,z1 \\
&+ 2\,C1^2\,a^2\,z2^2\,z1 - C1^2\,a\,z2^2 + 3\,C1^2\,a\,z1^2\,z2 - 3\,C1^2\,a^2\,z1^2\,z2 \\
&+ C1^2\,a^3\,z1^2\,z2 - 3\,C1^2\,z1^2\,z2^2\,a + 3\,C1^2\,z1^2\,z2^2\,a^2 \\
&- C1^2\,a^3\,z1^2\,z2^2 + z2^2\,C1^2 - 5\,z2\,C1^2 + 2\,z2^2\,C1^2\,z1 - z2\,C1^2\,z1^2 \\
&+ (-C1\,C2\,z2\,z1 + C2\,z2^2\,z1^2\,C1 + C1\,C2\,z2 - C2\,z2^2\,C1\,z1)\,\_Z^3 \\
&+ (C2\,C1\,a - 2\,C1\,C2 - C1^2\,a + C2^2 - 2\,C2^2\,z1 - C2^2\,z2 + C2^2\,z1^2 \\
&+ C1^2 + C1^2\,a\,z2 + 2\,C2^2\,z2\,z1 - C2^2\,z1^2\,z2 + 2\,C1\,C2\,z1 \\
&- 4\,C1\,C2\,z2 - C1^2\,z1\,z2 + C1^2\,a\,z1\,z2 - 3\,C2\,z1^2\,C1\,z2 \\
&+ C2\,z1^2\,C1\,a\,z2 - C2\,z1\,C1\,a - 2\,C2\,z1\,C1\,a\,z2 - C2\,z2^2\,z1\,C1\,a \\
&+ 7\,C1\,C2\,z2\,z1 + C2\,C1\,a\,z2 + C2\,z2^2\,C1\,z1 - 2\,C2\,z2^2\,z1^2\,C1 \\
&+ C2\,z2^2\,z1^2\,C1\,a - C1^2\,a\,z2^2\,z1 - z2\,C1^2 + z2^2\,C1^2\,z1 \\
&+ C2\,z2^2\,C1)\,\_Z^2 + (-4\,C2\,C1\,a - C1^2\,z1^2\,z2^2 + 6\,C1\,C2 \\
&+ 5\,C1^2\,a - C1^2\,a^2 - 2\,C2^2 + 4\,C2^2\,z1 + 2\,C2^2\,z2 - 2\,C2^2\,z1^2 \\
&- 4\,C1^2 - 6\,C1^2\,a\,z2 - 4\,C2^2\,z2\,z1 + 2\,C2^2\,z1^2\,z2 + C1^2\,a^2\,z2 \\
&- 6\,C1\,C2\,z1 + 3\,C1\,C2\,z2 + C2\,C1\,a^2 + C1^2\,a\,z1 + 2\,C1^2\,z1\,z2 \\
&- C1^2\,a^2\,z1 - 4\,C1^2\,a\,z1\,z2 + 2\,C1^2\,a^2\,z1\,z2 + C2\,z1^2\,C1\,a
\end{aligned}
$$

$$+ 7\, C2\, z1^2\, C1\, z2 - 6\, C2\, z1^2\, C1\, a\, z2 + C2\, z1^2\, C1\, a^2\, z2$$
$$+ 3\, C2\, z1\, C1\, a - C2\, z1\, C1\, a^2 + 8\, C2\, z1\, C1\, a\, z2 - C1\, a^2\, C2\, z2\, z1$$
$$+ C2\, z2^2\, z1\, C1\, a - 10\, C1\, C2\, z2\, z1 - 2\, C2\, C1\, a\, z2 + C2\, z2^2\, z1^2\, C1$$
$$- C2\, z2^2\, z1^2\, C1\, a + 3\, C1^2\, a\, z2^2\, z1 - C1^2\, a^2\, z2^2\, z1 + C1^2\, a\, z2^2$$
$$- 2\, C1^2\, a\, z1^2\, z2 + C1^2\, a^2\, z1^2\, z2 + 2\, C1^2\, z1^2\, z2^2\, a - C1^2\, z1^2\, z2^2\, a^2$$
$$- z2^2\, C1^2 + 5\, z2\, C1^2 - 2\, z2^2\, C1^2\, z1 + z2\, C1^2\, z1^2 - C2\, z2^2\, C1)\_Z)$$

contour1-contour2;

$$0$$

i.e., contours 1 and 2 are identically the same, so the boundary is that contour. What this shows is that there are no regions for which we have not found the optimal policy. Now we can calculate the contour equations for the simple case, by setting $\alpha_1 = \alpha_2$ and $C_1 = C_2$

z1:=z2;

$$z1 := z2$$

C1:=C2;

$$C1 := C2$$

contour1;

$$\mathrm{RootOf}((z2 + z2^2)\_Z^3 + (a\, z2 - 2\, z2^2 + a\, z2^2 - 6\, z2)\_Z^2$$
$$+ (-a^2\, z2^2 - 2\, a\, z2 + a\, z2^2 + 8\, z2 - a^2\, z2 + a)\_Z - 3\, a\, z2^2$$
$$+ 3\, a^2\, z2^2 - 2\, z2 + z2^2 + a^2 - a^3\, z2^2 + 4\, a^2\, z2 - 2\, a\, z2 - a^3\, z2 + 1$$
$$- 3\, a)$$

B.2 Boundary between Regions 2 and 3

The calculations involved in this case are almost identical to those of section 1. It is worth noting the reversal of a and b in the simple case.

First, we need to calculate the values of $P2$, $P3$ and P_1.

A1:=-C1/(1-b);

$$A1 := -\frac{C1}{1-b}$$

B1:=C1/(1-b)*(2-b-a*z1)/(1-z1);

$$B1 := \frac{C1\,(2-b-a\,z1)}{(1-b)\,(1-z1)}$$

A2:=C2/(1-a);

$$A2 := \frac{C2}{1-a}$$

B2:=C2/(1-a)*(1-a+z2*(1-b))/(1-z2);

$$B2 := \frac{C2\,(1-a+z2\,(1-b))}{(1-a)\,(1-z2)}$$

P3:=(B2-B1)/(A1-A2);

$$P3 := \frac{\dfrac{C2\,(1-a+z2\,(1-b))}{(1-a)\,(1-z2)} - \dfrac{C1\,(2-b-a\,z1)}{(1-b)\,(1-z1)}}{-\dfrac{C1}{1-b} - \dfrac{C2}{1-a}}$$

simplify(P3);

$$-(-2\,C1\,z2 - C2 + C1\,a^2\,z1 - C1\,a^2\,z1\,z2 + C1\,a\,z1\,z2 - C1\,a\,z1$$
$$- C1\,a\,b\,z2 + C1\,a\,b + C1\,z2\,b + 2\,C1\,a\,z2 + C2\,z2\,b^2\,z1$$
$$- 2\,C2\,z2\,b\,z1 + C2\,a\,b\,z1 + 2\,C1 + C2\,a + C2\,b + C2\,z1$$

$$- C2\, b\, z1 - C2\, z2\, b^2 + 2\, C2\, z2\, b + C2\, z2\, z1 - C2\, a\, b - C2\, a\, z1$$
$$- C2\, z2 - 2\, C1\, a - C1\, b) \Big/ ((-C1 + C1\, a - C2 + C2\, b)(-1 + z1)$$
$$(-1 + z2))$$

P2:=((1-z2)*(1-b)+C1/C2*(1-z2)*(1-a))/(2-z1-z2-(a+b-1)*(1-z1*z2));

$$P2 := \frac{(1 - z2)(1 - b) + \dfrac{C1\,(1 - z2)(1 - a)}{C2}}{2 - z1 - z2 - (a + b - 1)(1 - z2\, z1)}$$

P_1:=(2-z1*a-b-sqrt((z1*a+b)^2-4*z1*(a+b-1)))/(2*(1-z1));

$$P\_1 := \left(2 - a\, z1 - b - \sqrt{a^2\, z1^2 + 2\, a\, b\, z1 + b^2 - 4\, a\, z1 - 4\, b\, z1 + 4\, z1}\right) \Big/ ($$
$$2 - 2\, z1)$$

Now, we check that the contours along which P2=P_1 and P3=P_2 are the same.

contour1:=solve(P3=P_1,a);

$$contour1 := \mathrm{RootOf}(4\, C1\, z2\, C2 - 2\, C2^2\, z2\, z1 + C1^2\, z2^2 + 4\, C2^2$$
$$+ 2\, C2^2\, z1^2\, z2 + C2^2\, z2^2\, z1^2 + 4\, C1\, z2\, C2\, z1 - 4\, C1\, z2^2\, C2\, z1$$
$$- 2\, C1^2\, z2 - 4\, C2\, C1 - 5\, C2^2\, z1 + C1^2 + C2^2\, z1^2 - z1\, C1^2\, z2^2$$
$$- z1\, C2^2\, z2^2 + 2\, z1\, C1^2\, z2 - z1\, C1^2 + (2\, C1\, z2\, C2 - C2^2\, z2\, z1$$
$$+ C1^2\, z2^2 + C2^2 + C2^2\, z1^2\, z2 + 7\, C1\, z2\, C2\, z1 - 3\, C1\, z2^2\, C2\, z1$$
$$- 2\, C1^2\, z2 - 2\, C2\, C1 - C2^2\, z1 + C1^2 - z1\, C1^2\, z2^2 + 2\, z1\, C1^2\, z2$$
$$- z1\, C1^2 - 4\, C1\, C2\, z1 - C1\, z1^2\, z2\, C2\, b + C1\, z1^2\, z2^2\, C2\, b$$
$$+ C1\, z1^2\, z2\, C2 + C1\, z1^2\, C2 - 2\, C1\, z1^2\, z2^2\, C2 + C1\, C2\, b\, z1$$
$$- C2^2\, z1^2\, z2\, b - C2\, C1\, z2\, b + C2^2\, b\, z1 + C2\, C1\, b$$
$$- 2\, C1\, z2\, b\, C2\, z1 + C1\, z2^2\, b\, C2\, z1 + C2^2\, z2\, b\, z1 - C2^2\, b)\_Z^2 + ($$
$$-6\, C1\, z2\, C2 + 2\, C2^2\, z2\, z1 - 2\, C1^2\, z2^2 - 4\, C2^2 - 2\, C2^2\, z1^2\, z2$$

$$- C2^2\, z2^2\, z1^2 - 10\, C1\, z2\, C2\, z1 + 7\, C1\, z2^2\, C2\, z1 + 4\, C1^2\, z2$$
$$+ 6\, C2\, C1 + 5\, C2^2\, z1 - 2\, C1^2 - C2^2\, z1^2 + 2\, z1\, C1^2\, z2^2$$
$$+ z1\, C2^2\, z2^2 - 4\, z1\, C1^2\, z2 + 2\, z1\, C1^2 + 3\, C1\, C2\, z1$$
$$+ C1\, z1^2\, z2\, C2\, b - C1\, z1^2\, z2^2\, C2\, b - C1\, z1^2\, C2 + C1\, z1^2\, z2^2\, C2$$
$$- 2\, C1\, C2\, b\, z1 + 2\, C2^2\, z2^2\, z1^2\, b - C2^2\, z2^2\, z1^2\, b^2 + 3\, C2^2\, z1^2\, z2\, b$$
$$- C2^2\, z1^2\, z2\, b^2 - C1\, z2\, b^2\, C2\, z1 + 3\, C2\, C1\, z2\, b - C2\, z2\, b^2\, C1$$
$$- 6\, C2^2\, b\, z1 + C2^2\, z1^2\, b + C2^2\, z1\, b^2 - 4\, C2\, C1\, b + C1\, C2\, b^2$$
$$+ 8\, C1\, z2\, b\, C2\, z1 - 6\, C1\, z2^2\, b\, C2\, z1 - 4\, C2^2\, z2\, b\, z1$$
$$- 2\, C2^2\, z2^2\, b\, z1 + 2\, C2^2\, z2\, b^2\, z1 + C2^2\, z2^2\, b^2\, z1$$
$$+ C1\, z2^2\, C2\, b^2\, z1 + C1\, z2^2\, C2\, b + 5\, C2^2\, b - C2^2\, b^2 + C2^2\, z2\, b$$
$$- C2^2\, z2\, b^2)\_Z$$
$$+ (-C1\, z1^2\, z2\, C2 - C1\, z2\, C2\, z1 + C1\, C2\, z1 + C1\, z1^2\, z2^2\, C2)\_Z^3$$
$$+ C1\, C2\, b\, z1 - 3\, C2^2\, z2^2\, z1^2\, b + 3\, C2^2\, z2^2\, z1^2\, b^2 - 4\, C2^2\, z1^2\, z2\, b$$
$$+ 2\, C2^2\, z1^2\, z2\, b^2 + C1\, z2\, b^2\, C2\, z1 + C2^2\, z2^2\, b^3\, z1 + C2^2\, z2\, b^3\, z1$$
$$- 2\, C2\, C1\, z2\, b + C2\, z2\, b^2\, C1 + 7\, C2^2\, b\, z1 - C2^2\, z2\, b^3 - C2^2\, z1^2\, b$$
$$- 2\, C2^2\, z1\, b^2 + 3\, C2\, C1\, b - C1\, C2\, b^2 - 6\, C1\, z2\, b\, C2\, z1$$
$$+ 5\, C1\, z2^2\, b\, C2\, z1 + 6\, C2^2\, z2\, b\, z1 + 3\, C2^2\, z2^2\, b\, z1$$
$$- 5\, C2^2\, z2\, b^2\, z1 - 3\, C2^2\, z2^2\, b^2\, z1 - C1\, z2^2\, C2\, b^2\, z1 - C1\, z2^2\, C2\, b$$
$$- 6\, C2^2\, b + 2\, C2^2\, b^2 - 2\, C2^2\, z2\, b + 3\, C2^2\, z2\, b^2 - C2^2\, z2^2\, b^3\, z1^2)$$

contour2:=solve(P2=P$_1$,a);

$$contour2 := \mathrm{RootOf}(4\, C1\, z2\, C2 - 2\, C2^2\, z2\, z1 + C1^2\, z2^2 + 4\, C2^2$$
$$+ 2\, C2^2\, z1^2\, z2 + C2^2\, z2^2\, z1^2 + 4\, C1\, z2\, C2\, z1 - 4\, C1\, z2^2\, C2\, z1$$
$$- 2\, C1^2\, z2 - 4\, C2\, C1 - 5\, C2^2\, z1 + C1^2 + C2^2\, z1^2 - z1\, C1^2\, z2^2$$

$$- z1\ C2^2\ z2^2 + 2\ z1\ C1^2\ z2 - z1\ C1^2 + (2\ C1\ z2\ C2 - C2^2\ z2\ z1$$

$$+ C1^2\ z2^2 + C2^2 + C2^2\ z1^2\ z2 + 7\ C1\ z2\ C2\ z1 - 3\ C1\ z2^2\ C2\ z1$$

$$- 2\ C1^2\ z2 - 2\ C2\ C1 - C2^2\ z1 + C1^2 - z1\ C1^2\ z2^2 + 2\ z1\ C1^2\ z2$$

$$- z1\ C1^2 - 4\ C1\ C2\ z1 - C1\ z1^2\ z2\ C2\ b + C1\ z1^2\ z2^2\ C2\ b$$

$$+ C1\ z1^2\ z2\ C2 + C1\ z1^2\ C2 - 2\ C1\ z1^2\ z2^2\ C2 + C1\ C2\ b\ z1$$

$$- C2^2\ z1^2\ z2\ b - C2\ C1\ z2\ b + C2^2\ b\ z1 + C2\ C1\ b$$

$$- 2\ C1\ z2\ b\ C2\ z1 + C1\ z2^2\ b\ C2\ z1 + C2^2\ z2\ b\ z1 - C2^2\ b)\_Z^2 + ($$

$$-6\ C1\ z2\ C2 + 2\ C2^2\ z2\ z1 - 2\ C1^2\ z2^2 - 4\ C2^2 - 2\ C2^2\ z1^2\ z2$$

$$- C2^2\ z2^2\ z1^2 - 10\ C1\ z2\ C2\ z1 + 7\ C1\ z2^2\ C2\ z1 + 4\ C1^2\ z2$$

$$+ 6\ C2\ C1 + 5\ C2^2\ z1 - 2\ C1^2 - C2^2\ z1^2 + 2\ z1\ C1^2\ z2^2$$

$$+ z1\ C2^2\ z2^2 - 4\ z1\ C1^2\ z2 + 2\ z1\ C1^2 + 3\ C1\ C2\ z1$$

$$+ C1\ z1^2\ z2\ C2\ b - C1\ z1^2\ z2^2\ C2\ b - C1\ z1^2\ C2 + C1\ z1^2\ z2^2\ C2$$

$$- 2\ C1\ C2\ b\ z1 + 2\ C2^2\ z2^2\ z1^2\ b - C2^2\ z2^2\ z1^2\ b^2 + 3\ C2^2\ z1^2\ z2\ b$$

$$- C2^2\ z1^2\ z2\ b^2 - C1\ z2\ b^2\ C2\ z1 + 3\ C2\ C1\ z2\ b - C2\ z2\ b^2\ C1$$

$$- 6\ C2^2\ b\ z1 + C2^2\ z1^2\ b + C2^2\ z1\ b^2 - 4\ C2\ C1\ b + C1\ C2\ b^2$$

$$+ 8\ C1\ z2\ b\ C2\ z1 - 6\ C1\ z2^2\ b\ C2\ z1 - 4\ C2^2\ z2\ b\ z1$$

$$- 2\ C2^2\ z2^2\ b\ z1 + 2\ C2^2\ z2\ b^2\ z1 + C2^2\ z2^2\ b^2\ z1$$

$$+ C1\ z2^2\ C2\ b^2\ z1 + C1\ z2^2\ C2\ b + 5\ C2^2\ b - C2^2\ b^2 + C2^2\ z2\ b$$

$$- C2^2\ z2\ b^2)\_Z$$

$$+ (-C1\ z1^2\ z2\ C2 - C1\ z2\ C2\ z1 + C1\ C2\ z1 + C1\ z1^2\ z2^2\ C2)\_Z^3$$

$$+ C1\ C2\ b\ z1 - 3\ C2^2\ z2^2\ z1^2\ b + 3\ C2^2\ z2^2\ z1^2\ b^2 - 4\ C2^2\ z1^2\ z2\ b$$

$$+ 2\ C2^2\ z1^2\ z2\ b^2 + C1\ z2\ b^2\ C2\ z1 + C2^2\ z2^2\ b^3\ z1 + C2^2\ z2\ b^3\ z1$$

$$- 2\ C2\ C1\ z2\ b + C2\ z2\ b^2\ C1 + 7\ C2^2\ b\ z1 - C2^2\ z2\ b^3 - C2^2\ z1^2\ b$$

$$- 2\ C2^2\ z1\ b^2 + 3\ C2\ C1\ b - C1\ C2\ b^2 - 6\ C1\ z2\ b\ C2\ z1$$

$$+ 5\, C1\, z2^2\, b\, C2\, z1 + 6\, C2^2\, z2\, b\, z1 + 3\, C2^2\, z2^2\, b\, z1$$
$$- 5\, C2^2\, z2\, b^2\, z1 - 3\, C2^2\, z2^2\, b^2\, z1 - C1\, z2^2\, C2\, b^2\, z1 - C1\, z2^2\, C2\, b$$
$$- 6\, C2^2\, b + 2\, C2^2\, b^2 - 2\, C2^2\, z2\, b + 3\, C2^2\, z2\, b^2 - C2^2\, z2^2\, b^3\, z1^2)$$

contour1-contour2;

$$0$$

Now, again we can calculate the contour equations for the simple case.

z1:=z2;

$$z1 := z2$$

C1:=C2;

$$C1 := C2$$

contour1;

$$\mathrm{RootOf}((z2^2 + z2)\, \_Z^3 + (-6\, z2 - 2\, z2^2 + z2\, b + z2^2\, b)\, \_Z^2$$
$$+ (8\, z2 - 2\, z2\, b + z2^2\, b - z2^2\, b^2 - z2\, b^2 + b)\, \_Z + 3\, z2^2\, b^2 - 3\, z2^2\, b$$
$$- z2^2\, b^3 - z2\, b^3 - 2\, z2 + 1 + b^2 + z2^2 - 3\, b - 2\, z2\, b + 4\, z2\, b^2)$$

Bibliography

[1] Ahlswede, R. and Wegener, I. (1979) *Search Problems,* John Wiley, Chichester.

[2] Assaf, D. and Sharlin-Bilitzky, A. (1994) Dynamic Search for a Moving Target *J. Appl. Prob.* **31** pp 438-457.

[3] Benkoski, S.J., Monticino, M.G. and Weisinger, J.R. (1991) A Survey of the Search Theory Literature *Nav. Res. Logistics* **38**, pp 469–494.

[4] Gal, S. (1980) *Search Games,* Academic Press, San Francisco.

[5] MacPhee, I.M. and Jordan, B.P.(1995) Optimal Search for a Moving Target *Probability in Engineering and the Informational Sciences* **9**, pp 159-182.

[6] Kan, Y.C. (1977) Optimal Search of a Moving Target *Operations Research* **25**, pp 864–870.

[7] Koopman, B.O. (1980) *Search and Screening : General Principles with Historical Applications,* Pergamon, New York.

[8] Nakai, T. (1973) A Model of Search for a Target Among Three Boxes: Some Special Cases *J. Oper. Res. Soc. of Japan* **16**, pp 151–162.

[9] Nakai, T. (1980) A Search Model for a Moving Target in which the Decision Wait is Permitted *Math. Japonica* **25**, pp 597–605.

[10] Penrose, M.D. (1995) Personal Communication.

[11] Pollock, S.M. (1970) A Simple Model of Search for a Moving Target *Operat. Res.* **18**, pp 883–903.

[12] Ross, S.M. (1970) *Applied Probability Models with Optimization Examples*, Holden-Day, San Francisco.

[13] Ross, S.M. (1983) *Introduction to Stochastic Dynamic Programming*, Academic Press, New York.

[14] Stone, L.D. (1975) *Theory of Optimal Search*, Academic Press, New York.

[15] Strauch, R. (1966) Negative Dynamic Programming *Ann. Math. Statist.* **37**, pp 871-890.

[16] Thomas, L.C. and Eagle, J.N. (1995) Criteria and Approximate Methods for Path-Constrained Moving-Target Search Problems *Nav. Res. Logistics* **42**, pp 27-38.

[17] Washburn, A.R. (1983) Search for a Moving Target : The FAB Algorithm *Operations Research* **31**, pp 739-751.

[18] Weber, R.R. (1986) Optimal Search for a Randomly Moving Object, *J. Appl. Prob.* **23**, pp 708–717.

[19] Weber, R.R. (1994) Personal Communication.

[20] Whittle, P. (1984) *Optimisation Over Time, Dynamic Programming and Stochastic Control, Volume II*, John Wiley, Chichester.