# LINEAR SEARCH PROBLEM WITH LOW SENSING ON TWO RAYS

BY

ARGEN M. WEST

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mathematics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Professor Professor Lee DeVille, Chair
Professor Vadim Zharnitsky, Director of Research
Professor Jared Bronski
Associate Professor Zoi Rapti

# Abstract

We consider a generalization of the linear search problem where the searcher has low sensing capabilities on two rays. We first show the necessary conditions for an optimal search plan to exist. We then investigate properties of optimal search plans and show that optimal search plans are defined by an underlying fourth order recurrence relation.

We then develop numerical methods that aid in estimating and finding optimal search plans. In Chapter 4, we present an algorithm that produces a search plan that approximates the minimum expected cost up to any desired accuracy for any probability density distribution. In Chapter 5, for specific distributions, properties of the underlying dynamics are used to numerically find optimal search plans.

*To my parents for their support and encouragement*

# Acknowledgments

I would like to thoroughly thank my adviser, Vadim Zharnitsky, for his guidance and insight in this research. I express my gratitude for my preliminary and doctoral committee members, Yuliy Baryshnikov, Jared Bronski, Lee DeVille and Zoi Rapti.

I would like to thank my undergraduate adviser Aghalaya S. Vatsala. If not for her, I would not have ended up at the University of Illinois. I'm also appreciative of my other professors from my undergraduate years, Patricia Beaulieu and Victor P. Schneider, for taking their time outside of class to mentor me and write multiple letters of recommendation.

I am grateful to my parents, Noah and Thelma West, for their moral and financial support throughout my time as an undergraduate and graduate student. I am thankful to Kathryn L. Gainey for her endless support and encouragement. I would like to thank all the close friends for all the good memories during my tenure of University of Illinois, especially Bill Karr, Rob Madsen and Joe Nance.

I'm eternally appreciative to my advisers during my internships at John Deere and Dow Agro Sciences, Kevin Armstrong and Steve Cryer respectively. I owe many thanks to them for helping me begin my career as a professional mathematician.

Lastly, I owe much gratitude to the National Science Foundation for their financial support from their grants DMS 1345032 "MCTP: PI4: Program for Interdisciplinary and Industrial Internships at Illinois" and DMS 08-38434 "EMSW21-MCTP: Research Experience for Graduate Students."

# Table of Contents

# Chapter 1

# Introduction

## 1.1 Background and Motivation

The linear search problem was first presented by R. Bellman in 1963 [6] and studied by A. Beck and W. Franck [2, 10]. The linear search problem is as follows. Suppose an object, $\boldsymbol{l}$, is located somewhere on the real line according to a known probability distribution function, $p(x)$. A searcher or robot starts at the origin but does not know the where the object is located. It searches for the object by following a sequence of turning points or excursions, $\boldsymbol{x} = \{x_i\}_{i=1}^{\infty}$, called a search plan, with the turning points satisfying:

$$... \leq -x_6 \leq -x_4 \leq -x_2 \leq 0 \leq x_1 \leq x_3 \leq x_5 \leq ...$$

The searcher will follow the search plan until it locates the object. The total distance traveled by the searcher is the cost. The main goal of the linear search problem is to find an optimal search plan that minimizes the expected cost.

Suppose the probability distribution function, $p(x)$, is symmetric about the origin and let $f(x) = P(\boldsymbol{l} > |x|)$. Then the cost function is given by $L(f) + 2E(\boldsymbol{x}, f)$ [4], where,

$$L(f) = \int_{-\infty}^{\infty} f(x)dx \text{ and } E(\boldsymbol{x}, f) = \sum_{n=1}^{\infty} x_n(f(x_n) + f(x_{n-1})).$$

If the probability distribution function satisfies certain conditions, an optimal search plan that yields the minimum expected cost will exist [2, 10]. However, finding the optimal search plan and the minimum expected cost are challenging. When an optimal search plan exists, there is an underlying second order recurrence relation that the optimal search plan must satisfy. If the probability distribution function is symmetric about the origin, then the recurrence is given by:

$$(x_{n+1} + x_n)f'(x_n) + f(x_n) + f(x_{n-1}) = 0.$$

Since $x_0$ is taken to be zero, $x_2$ is defined by the first excursion, $x_1$. Since $x_3$ depends on $x_2$ and $x_1$, $x_3$ is

also solely defined by the first excursion, $x_1$. Hence, all $x_n$ depend solely on the choice of $x_1$. Thus, the expected cost function is reduced to a one variable function. Although, the expected cost function is a one variable function, to the author's knowledge, there have been no analytical or efficient numerical methods to find an optimal $x_1$ which yields the minimum expected cost for a general distribution. However, for the normal distribution, A. Beck, M. Beck [4, 5] and P. Rousseeuw [11] used a combination of numerical and analytical estimates to find the optimal initial turning point.

Although the linear search problem was first posed in the 60s by Bellman [6], it has been of continual interest to researchers, not only because of its applications but also because its intriguing underlying mathematics. Alpern, Gal, Beck and Newman have investigated the linear search problem from a game theoretic perspective, in which one wishes to minimize the expected cost in a worst case scenario [1, 3]. More recently, De Pablo et al. studied the linear search problem in connection with a robotic searcher [9]. Another recent study of the linear search problem was the search on one ray where the searcher is "blind" or has low sensing capabilities [13, 14]. Baryshnikov et al. first studied the blind search due to its applications in searching in unstructured Peer-to-Peer storage structures [13]. A blind searcher can not detect the object when it comes across it. It must bring the object back to the origin to verify if the object has been collected. If it is determined that the object was not found the searcher does another excursion. This process is repeated until the object is found. Another feasible application of the linear search problem with low sensing is unmanned aerial vehicles. If an unmanned aerial vehicle was trying avoid detectable communication, it would not be able to relay information while out on a survey or mission. Hence, it would have to return to base to be able to relay information from its mission.

Originally, the goal of this research was to study the blind linear search problem on multiple rays. However, it soon became clear that this would be an arduous task, as the two ray case itself presented a plethora of challenging problems. So, the focus of this research shifted to the two ray case. As the one ray case naturally shed mathematical light on the two ray case, one would expect the two ray case to shed light on the multiple ray case. For example, when there are multiple rays there is no reason a priori that optimal search plans should be monotone or alternating between rays. As we will see later, these questions will be addressed for the two ray case. To allow for a cohesive and complete story of the two ray case, we first give a brief overview of the one ray case.

## 1.2 Overview of the One Ray Case

Baryshnikov and Zharnitsky did an in depth study of blind linear search problem on one ray in their paper [14]. The search on one ray is as follows. Suppose an object, $\boldsymbol{l}$, is hidden somewhere on the positive ray, $\mathbb{R}_+$, according to some known probability distribution, $p(x)$. A searcher with low sensing follows a sequence of excursions. It is assumed the searcher cannot detect the object when it comes across it and must bring the object to the origin for verification. After each excursion, the searcher returns to the origin to determine if the object has been discovered or not. Once the object is located the search is complete. Hence, the total distance traveled, or total cost, is a random variable. If $\boldsymbol{x}$ is a search plan with turning points satisfying $0 = x_0 < x_1 < x_2 < ...$ and $x_i \to \infty$ then the expected cost function for the search plan is given by:

$$E(\boldsymbol{x}) = \sum_{i=1}^{\infty} x_i P(\boldsymbol{l} > x_{i-1}) = \sum_{i=1}^{\infty} x_i f(x_{i-1}).$$

As with the original search problem, there is an underlying recurrence relation that defines optimal search plans for $E(\boldsymbol{x})$. Since $E(\boldsymbol{x})$ depends on $x_n$ via only the term $f(x_{n-1})x_n + f(x_n)x_{n+1}$, we can differentiate $E(\boldsymbol{x})$ with respect to $x_n$. Differentiating $E(\boldsymbol{x})$ with respect to $x_n$ yields the following second order recurrence relation:

$$f(x_{n-1}) + x_{n+1}f'(x_n) = 0.$$

Since $x_0$ is taken to be zero, the choice of $x_1$ defines the whole sequence. Baryshnikov and Zharnitsky used properties of the underlying dynamics to find the optimal initial turning point for the exponential distribution. For $f(x) = \exp(-x)$, it was analytically shown that under the appropriate coordinate system, the phase space of trajectories from the recurrence relation was split into two regions by a separatrix. The separatrix was an invariant curve in the phase space that separated the region of points that generated monotone sequences from the points that did not generate monotone sequences.

**Definition 1.2.1.** *The region $M_k$ of k-step monotonicity is defined as collection of points in the phase space such that k-fold application of the R produces a monotonic (along x coordinate) sequence. The intersection of all $M_k$ is denoted by $M_\infty := \cap M_k$ and is called the region of monotonicity. Its complement is called the chaotic region. [14]*

Points in the region of monotonicity not on the separatrix generated sequences that grew too rapidly to yield an optimal search plan. However, it was shown that candidates for optimal $x_1$ lay on the separatrix. Let $f(x) = \exp(-x)$. Then the recurrence is given by: $x_{n+1} = \exp(x_n - x_{n-1})$. The recurrence can be given

by the following change of coordinates:

$$\begin{aligned} x_{n+1} &= \exp(y_n) \\ y_{n+1} &= x_{n+1} - x_n = \exp(y_n) - x_n \end{aligned} \tag{1.1}$$

The inverse for the change of coordinates is given by:

$$\begin{aligned} x_n &= x_{n+1} - y_{n+1} \\ y_n &= \ln(x_{n+1}) \end{aligned} \tag{1.2}$$

In the phase space, the mapping of consecutive trajectories, $(x_n, y_n)$, is given by:

$$\mathbf{R} : (x, y) \mapsto (X, Y) = (\exp(y), \exp(y) - x).$$

The inverse of the mapping is given by:

$$\mathbf{R}^{-1} : (X, Y) \mapsto (x, y) = (X - Y, \ln(Y)).$$

Baryshnikov and Zharnitsky analytically proved that a separatrix existed and that backwards iterations on the boundary of $y > 0$ converged to the separatrix (See Figure 1.1).

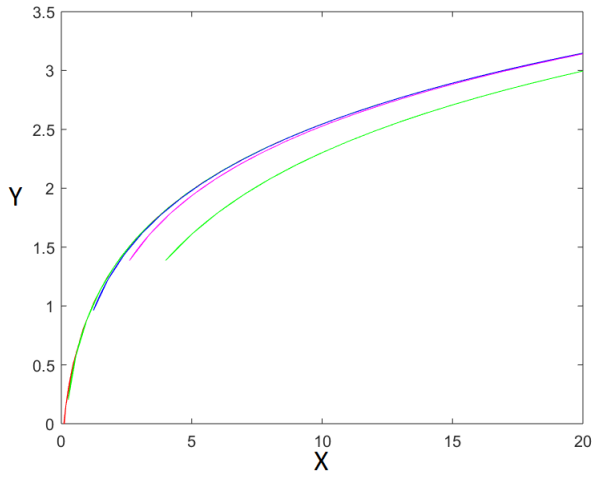


Figure 1.1: Iterations under (1.2) that started on the boundary of $y > 0$ converging to the separatrix.

Since $x_0 = 0$, we have the initial condition: $y_1 = x_1 - x_0 = x_1$. In the phase space, the line $y = x$ represents this initial condition. It was shown that for $x_1$ to be optimal, it must be in the intersection of

the initial points and the separatrix. In Figure 1.2 two points from the line of initial points intersect the separatrix. However, numerically, one yielded a lower expected cost. For $f(x) = \exp(-x)$ it was shown that the separatrix, $\phi(x)$, was a smooth function of $x$ such that $\phi(x) = \ln(x) + O\left(\frac{\ln(x)}{x}\right)$. We will see later, that this fact will be immensely helpful in finding invariant structures of interest for the two ray case.

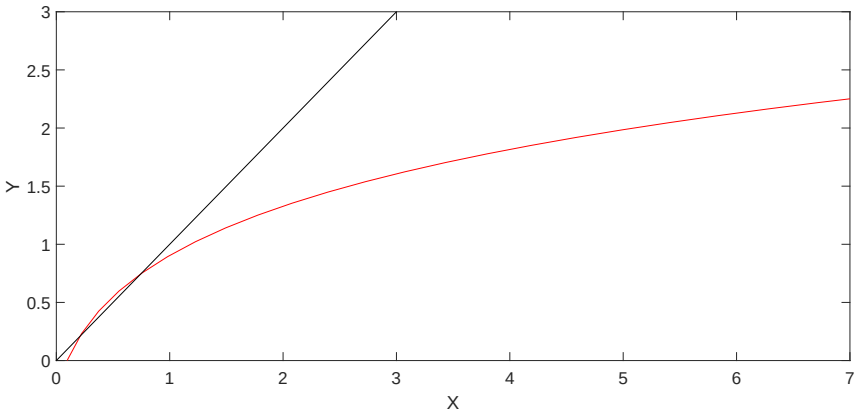


Figure 1.2: The straight line, $y = x$, corresponds to possible initial turning points. The other curve is the separatrix.

## 1.3 Objective and Outline of Results

The goal of this research is to investigate the linear search problem with a blind searcher on the real line or, equivalently, two rays with a symmetric probability distribution about the origin. A symmetric probability distribution is a reasonable assumption since there is no prior knowledge of which ray the hidden objected is located upon. As we will see later, an optimal sequence will be defined by a fourth order recurrence relation. Since the recurrence relation is of fourth order, the expected cost function will depend not only on $x_1$ but also on $x_2$. Finding the optimal $x_1$ has been a challenging task for the other variations of the linear search problem. Hence, one would expect finding the optimal $x_1$ and $x_2$ for the two ray case would present even more of a challenge. If we wish to be able to find $x_1$ and $x_2$ analytically or numerically, we must first determine the necessary conditions for an optimal sequence to exist.

Firstly, in Chapter 2, we prove some properties of an optimal sequence and show the necessary conditions for an optimal sequence to exist. One of the important findings was that optimal sequences must be monotone. This result is of great importance because it vastly reduces the number of sequences which one has to consider as candidates for optimal search plans. This is immensely helpful for numerically finding optimal search plans. For example, in Chapter 4, the number of computations in the dynamic programming

algorithm is greatly reduced since one does not have to consider non-monotone sequences. Monotonicity also implies that optimal search plans must be in the region of monotonicity. If monotonicity holds, then knowing whether or not a separatrix exists of high interest.

Next, we investigate if optimal search plans must alternate between rays. In the original linear search problem the searcher always alternates between rays on the real line. This is because the searcher can detect the object when it comes across it. However, it is not necessarily clear that for the blind linear search problem optimal search plans must alternate between rays. In Chapter 3 we give a simple example were an alternating search plan and a non-alternating search plan both yield the minimum expected cost. There would be huge ramifications if it turned out that optimal search plans could either be alternating or non-alternating. Firstly, how would one be able to determine which type of search plan yielded the minimum expected cost? Secondly, if optimal sequences were not alternating it would be nearly impossible to study and understand the underlying dynamics because the recurrence relation would no longer have a nice and manageable form. However, it turns out for unimodal and bimodal distributions that are symmetric about the origin, optimal search plans must be alternating. Which is significant, because this is a vast family of distributions containing many common distributions such as the normal, logistic, Laplace and student-t distributions.

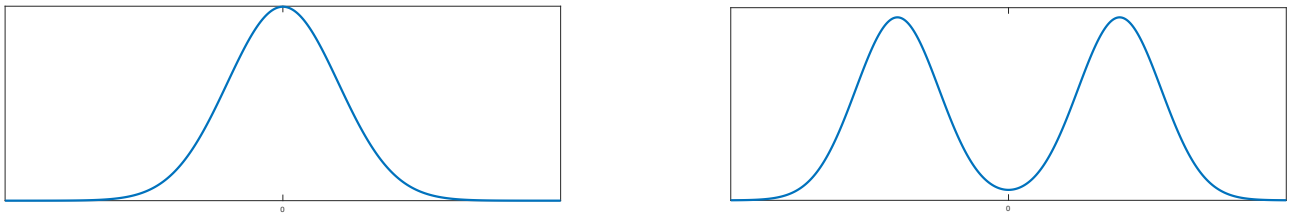

Figure 1.3: General shape of a unimodal and bimodal distribution

Although there have been no efficient methods for finding optimal $x_1$ for general distributions, there have been algorithms that generate sequences which estimate the minimum expected cost up to any accuracy [1]. In Chapter 4, using the fact that for a large class of distributions optimal sequences are monotone and alternating between rays, we are able to develop a dynamical programming algorithm for the blind linear search problem on the real line. This algorithm can estimate the expected minimum cost for any distribution up to any desired accuracy. An algorithm for the one ray case is also developed and given in the appendix. We then compare results from the algorithm to results from Chapter 5. Results from the dynamic program from the one and two ray cases generate $x_1$ and $x_2$ which are close in value to the optimal $x_1$ and $x_2$ generated from other methods. This is very intriguing because the dynamic programming does not use the

underlying recurrence relation. As stated before, there have been no efficient numerical methods to find an optimal $x_1$. Hence, to the author's knowledge, this is the first time that it has been noticed that a dynamic programming algorithm generates an initial turning point close to the optimal initial turning point. So, not only is this algorithm useful in finding sequences that estimate the expected minimum cost, it is also useful in validating other numerical methods.

For the one ray case with $f(x) = \exp(-x)$, optimal sequences are contained in an invariant curve in the phase space. In Chapter 5, using some numerics with careful estimates, we show that for the two ray case with $f(x) = .5\exp(-x)$ there is strong evidence that suggests there is an invariant curve in the phase space where optimal points lay. We then make a conjecture that with the Laplace distribution, optimal sequences are contained on a one dimensional invariant curve. At the end of Chapter 5, using the underlying recurrence relation, we present an ad hoc procedure that aids in numerically finding invariant structures where optimal initial turning points lay.

# Chapter 2

# Existence of an Optimal Search Plan

In this chapter we define the expected cost function for the blind linear search problem on two rays and give the necessary conditions for an optimal search plan to exist. We then give some properties of optimal search plans. Naturally, some properties from the one ray case extend to analogous properties for the two ray case. These properties include bounds on the minimum expected cost and the Lipschitz requirement on $f(x)$ for optimal sequences to exist. Monotonicity on the other hand, does extend naturally to the two ray case. Physically it is plausible that an optimal search plan must be monotone on one ray. It would not make sense to stop, turn around and head back to the origin in a region that has been previously searched. It can also be readily checked mathematically that monotonicity holds for the one ray case. For the two ray case, it is not necessarily clear, physically or mathematically, that optimal search plans must be monotone, i.e, excursions must always be increasing between rays. However, using several mathematical arguments we are able to prove monotonicity for optimal search plans. Then at the end of the chapter, we show optimal search plans must satisfy an underlying fourth order recurrence relation.

## 2.1 Distance Traveled and Expected Cost

Suppose an object, $\boldsymbol{l}$, is located somewhere on the real line according to a symmetric probability density function, $p(x)$. Suppose $p(x)$ is symmetric about the origin and let

$$c = \sup\{x | P(\boldsymbol{l} > x) > 0\}. \tag{2.1}$$

Let $\boldsymbol{x}$ be a search plan such that $\lim_{i\to\infty} x_i = c$ and has turning points satisfying:

$$... \leq -x_6 \leq -x_4 \leq -x_2 \leq 0 \leq x_1 \leq x_3 \leq x_5 \leq ...$$

Excursions, $x_{2n-1}$, correspond to excursions on the positive ray, $\mathbb{R}_+$, and $x_{2n}$, correspond to excursions on the negative ray, $\mathbb{R}_-$. If $c$ is finite, then $\boldsymbol{x} = \{x_i\}_{i=1}$ may or may not have a finite number of turning points

(See Proposition 2.1.3 and Example 3.0.1). Let $D(\boldsymbol{l}, \boldsymbol{x})$ be the total distance traveled when the object is located at $\boldsymbol{l}$ and the search plan is given by $\boldsymbol{x}$. If $\boldsymbol{l}$ is between $x_{n-2}$ and $x_n$, then the total distance traveled until the objected is collected and brought back to the origin is given by the random variable:

$$D(\boldsymbol{l}, \boldsymbol{x}) = 2\sum_{i=1}^{n} x_i.$$

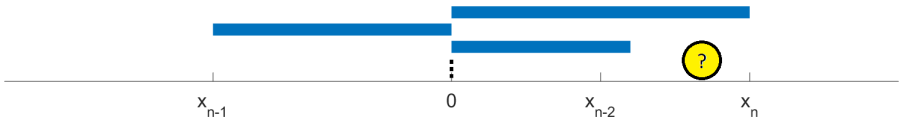


Figure 2.1: The hidden object is between $x_{n-2}$ and $x_n$ on the positive ray. Hence, the search would stop after excursion $x_n$.

The probability that $2\sum_{i=1}^{n} x_i$ is the total distance traveled is $P(x_{n-2} < \boldsymbol{l} < x_n)$. For a specific random variable $\boldsymbol{l}$, $\mathbb{E}(D(\boldsymbol{l}, \boldsymbol{x}))$ depends on the choice of $\boldsymbol{x}$. Define $E(\boldsymbol{x})$ as $2E(\boldsymbol{x}) = \mathbb{E}(D(\boldsymbol{l}, \boldsymbol{x}))$. Let $x_{-1}$ and $x_0$ be zero, then expected distance traveled is given by:

$$\mathbb{E}(D(\boldsymbol{l}, \boldsymbol{x})) = 2E(\boldsymbol{x}) = 2\sum_{n=1}^{\infty}\sum_{i=1}^{n} x_i P(x_{n-2} < \boldsymbol{l} < x_n).$$

Rearranging the terms, dividing by 2, and letting $f(x) = P(\boldsymbol{l} > |x|)$ yields:

$$E(\boldsymbol{x}) = \sum_{n=1}^{\infty} x_n(P(\boldsymbol{l} > x_{n-1}) + P(\boldsymbol{l} > x_{n-2})) = \sum_{n=1}^{\infty} x_n(f(x_{n-1}) + f(x_{n-2})). \tag{2.2}$$

We will see later, sometimes it can be useful to rearrange $E(\boldsymbol{x})$. Recall $x_{-1}$ and $x_0$ are taken to be zero. Rearranging terms in $E(\boldsymbol{x})$ yields:

$$\begin{aligned} E(\boldsymbol{x}) &= x_1(f(x_{-1}) + f(x_0)) + x_2(f(x_1) + f(x_0)) + x_3(f(x_2) + f(x_1)) + x_4(f(x_3) + f(x_2)) + ... \\ &= f(x_{-1})(0 + x_1) + f(x_0)(x_1 + x_2) + f(x_1)(x_2 + x_3) + f(x_2)(x_3 + x_4) + ... \\ &= f(x_{-1})(x_0 + x_1) + f(x_0)(x_1 + x_2) + f(x_1)(x_2 + x_3) + f(x_2)(x_3 + x_4) + ... \end{aligned}$$

Hence, $E(\boldsymbol{x})$ can be expressed as:

$$\sum_{n=-1}^{\infty} f(x_n)(x_{n+1} + x_{n+2}). \tag{2.3}$$

It is also useful to note the relation between $f(x)$ and the probability distribution, $p(x)$:

$$f'(x) = -p(x) \quad x \geq 0. \tag{2.4}$$

One of the main goals of a search problem is to find an optimal search plan that minimizes the expected cost. If we hope to be able to minimize the expected cost, we need $L < \infty$, where $L = \int_{-\infty}^{\infty} f(x)\,dx$. This is because, for any search plan, $\boldsymbol{x}$, $E(\boldsymbol{x}) \geq \frac{L}{2}$.

**Proposition 2.1.1.** $E(\boldsymbol{x}) \geq \frac{L}{2}$.

*Proof.* From the definition of $D(\boldsymbol{l}, \boldsymbol{x})$ we have that $D(\boldsymbol{l}, \boldsymbol{x}) \geq |\boldsymbol{l}|$. So, $2E(\boldsymbol{x}) = \mathbb{E}(D(\boldsymbol{l}, \boldsymbol{x})) \geq \mathbb{E}(|\boldsymbol{l}|) = L$. Thus, $E(\boldsymbol{x}) \geq \frac{L}{2}$. □

If $c$ is finite, then there exists a search plan, $\boldsymbol{x}$, such that $E(\boldsymbol{x}) \leq 1.5c$. This is because the expected cost of going to $c$ on the first ray then $c$ is on the second ray is $c + c(f(c) + .5) = c + c(0 + .5) = 1.5c$. If $c = \infty$ and the expected value of the location of the hidden object is bounded, then we can place an upper bound on the minimum expected cost.

**Proposition 2.1.2.** *Let $\epsilon > 0$. If $c = \infty$ and $L < \infty$, then there exists a search plan, $\boldsymbol{x}$, such that $E(\boldsymbol{x}) \leq 8L + \epsilon$.*

*Proof.* Let $C > 0$, $x_0 = 0$ and $x_i = C2^{i-1}$. Since $f(x)$ is decreasing we will use a right-hand Riemann sum to give a lower bound on the integral, $\int_0^{\infty} f(x)dx$.

$$\frac{L}{2} = \int_0^{\infty} f(x)dx \geq \sum_{i=1}^{\infty}(C2^i - C2^{i-1})f(C2^i) = \sum_{i=1}^{\infty} \frac{C2^i}{2} f(C2^i) = \sum_{i=1}^{\infty} \frac{C2^{i+2}}{8} f(C2^i).$$

So, we have:

$$\frac{L}{2} \geq \sum_{i=1}^{\infty} \frac{C2^{i+2}}{16}(f(C2^{i+1}) + f(C2^i)) = \sum_{i=1}^{\infty} \frac{x_{i+3}}{16}(f(x_{i+2}) + f(x_{i+1})).$$

Multiplying both sides by 16 then adding $x_1 = C$, $x_2 = 2C$ and $x_3 = 4C$ to both sides yields:

$$8L + C + 2C + 4C \geq C + 2C + 4C + \sum_{i=1}^{\infty} x_{i+3}(f(x_{i+1}) + f(x_{i+2})).$$

From the definition of $f(x) = P(\boldsymbol{l} > |x|)$ we have that $f(x) \leq .5$. Since $f(x) \leq .5$,

$$8L + C + 2C + 4C \geq C + 2C(f(C) + .5) + 4C(f(2C) + f(C)) + \sum_{i=1}^{\infty} x_{i+3}(f(x_{i+2}) + f(x_{i+1})) = E(\boldsymbol{x}).$$

Letting $C = \frac{\epsilon}{7}$ yields $8L + \epsilon \geq E(\boldsymbol{x})$. □

If $c = \infty$, then there are infinity many turning points in $\boldsymbol{x}$. However, if $c$ is finite then an optimal search plan may or may not have infinitely many turning points.

**Proposition 2.1.3.** *If $c$ is finite and $\lim_{x \to c^-} \frac{f(x)}{c - x} = 0$, then an optimal search plan must have infinitely many turning points.*

*Proof.* Suppose $f(x)$ has a finite support of $[-c, c]$ and suppose $\boldsymbol{x}$ is an optimal search plan with finitely many turning points. In the next section we will see optimal plans must be monotone. Hence, if the searcher goes to $c$ on one ray then the searcher must go to $c$ on the other ray. Let $\boldsymbol{x} = x_1, x_2, ..., x_{n-2}, c, c$ and $\boldsymbol{y} = x_1, x_2, ..., x_{n-1}, y, c, c$.

$$E(\boldsymbol{x}) = x_1 + ... + x_{n-2}(f(x_{n-3}) + f(x_{n-4})) + c(f(x_{n-2}) + f(x_{n-3})) + c(f(c) + f(x_{n-2}))$$

$$E(\boldsymbol{y}) = x_1 + ... + x_{n-2}(f(x_{n-3}) + f(x_{n-4})) + y(f(x_{n-2}) + f(x_{n-3})) + c(f(y) + f(x_{n-2})) + c(f(c) + f(y))$$

So, $E(\boldsymbol{x}) > E(\boldsymbol{y})$ if and only if $c(f(x_{n-2}) + f(x_{n-3})) - y(f(x_{n-2}) + f(x_{n-3})) - 2cf(y) > 0$. Which happens if and only if $f(x_{n-2}) + f(x_{n-3}) > \frac{2cf(y)}{c - y}$. Since $\lim_{y \to c^-} \frac{f(y)}{c - y} = 0$, such a $y$ can always be chosen. Hence, $\boldsymbol{x}$ is not an optimal sequence. □

## 2.2 Monotonicity of an Optimal Search Plan

For the expected cost for one ray case with a blind searcher it can be checked that removing all points of non-monotonicity in a search plan yields a lower expected cost. Hence, an optimal search plan will be monotone. However, for the two ray case it is not intuitively clear why a search plan should be monotone. For example, if the search plan was $\boldsymbol{x} = 2, 1, 5, 3, ...$, the searcher would be covering new ground on each of its first four excursions (See Figure 2.2). However, it turns out that an optimal search plan should be monotone. To be able to prove a search plan must be monotone, we will need the following propositions.

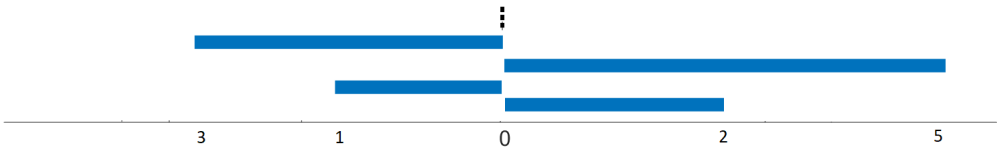


Figure 2.2: Let $\boldsymbol{x} = 2, 1, 5, 3, ...$ be the search plan. Then the searcher will cover new ground on each of its first four excursions. However, $\boldsymbol{x}$ is not monotone.

**Proposition 2.2.1.** *A sequence $\boldsymbol{x}$ with $x_n \leq x_{n-1}$ and $x_n \leq x_{n-2}$ for some $n$ is not optimal.*

*Proof.* Remove $x_n$ from $\boldsymbol{x}$ and call the new sequence $\hat{\boldsymbol{x}}$. $E(\boldsymbol{x})$ depends on $x_n$ via the terms $x_{n+1}f(x_n)$, $x_{n+2}f(x_n)$ and $x_n(f(x_{n-1}) + f(x_{n-2}))$. So, the difference $E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}})$ yields a finite sum. Subtracting $E(\hat{\boldsymbol{x}})$ from $E(\boldsymbol{x})$ and canceling out terms we find that

$$E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) = x_n(f(x_{n-1}) + f(x_{n-2})) + x_{n+1}(f(x_n) - f(x_{n-2})) + x_{n+2}(f(x_n) - f(x_{n-1})).$$

Since $f(x_n) \geq f(x_{n-1})$ and $f(x_n) \geq f(x_{n-2})$, $E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) > 0$. Hence, $E(\boldsymbol{x})$ is not optimal. □

Proposition 2.2.1 proved an excursion cannot be less than or equal to the previous two excursions. Proposition 2.2.2 will prove an excursion cannot be greater than or equal to the next two excursions.

**Proposition 2.2.2.** *A sequence $\boldsymbol{x}$ with $x_n \geq x_{n+1}$ and $x_n \geq x_{n+2}$ for some $n$ is not optimal.*

*Proof.* Remove $x_n$ from $\boldsymbol{x}$ and call the new sequence $\hat{\boldsymbol{x}}$. $E(\boldsymbol{x})$ depends on $x_n$ via the terms $x_{n+1}f(x_n)$, $x_{n+2}f(x_n)$ and $x_n(f(x_{n-1}) + f(x_{n-2}))$. So, the difference $E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}})$ yields a finite sum. Subtracting $E(\hat{\boldsymbol{x}})$ from $E(\boldsymbol{x})$ and canceling out terms we find that

$$E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) = f(x_n)(x_{n+1} + x_{n+2}) + x_n f(x_{n-2}) - x_{n+1}f(x_{n-2}) + x_n f(x_{n-1}) - x_{n+2}f(x_{n-1}).$$

Since $x_n \geq x_{n+1}$ and $x_n \geq x_{n+2}$, $E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) > 0$. Hence, $E(\boldsymbol{x})$ is not optimal. □

A priori, one would expect that an excursion on a ray should be greater than the previous excursion on the same ray. In more mathematical terms it should be the case that $x_n < x_{n+2}$. Using the previous two proposition we can prove this statement.

**Theorem 2.2.3.** *If a sequence $\boldsymbol{x}$ is optimal then it will have $x_n < x_{n+2}$.*

*Proof.* Let $\boldsymbol{x}$ be optimal. By Proposition 2.2.2 $x_n < x_{n+1}$ or $x_n < x_{n+2}$. If $x_n < x_{n+2}$, then we are done. Suppose $x_n \geq x_{n+2}$. Thus, $x_n < x_{n+1}$. Hence, $x_{n+2} \leq x_n < x_{n+1}$. However, $x_{n+2} \leq x_n$ and $x_{n+2} < x_{n+1}$ violates Proposition 2.2.1. So, we must have $x_n < x_{n+2}$. □

Finally using the propositions and theorem above we are able to show monotonicity holds for an optimal search plan when $f(x)$ is strictly decreasing.

**Theorem 2.2.4.** *If $f(x)$ is strictly decreasing then a sequence $\boldsymbol{x}$ with $x_n < x_{n-1}$ is not optimal.*

*Proof.* Let $\boldsymbol{x}$ be an optimal search plan and suppose $x_n < x_{n-1}$ is the first occurrence of non-monotonicity. So, we have $x_1 \leq x_2 \leq x_3 \leq ... \leq x_{n-4} \leq x_{n-3} \leq x_{n-2} \leq x_{n-1}$. If $x_n \leq x_{n-2}$ then $x_n$ would violate Proposition 2.2.1. So, we must have $x_{n-2} < x_n < x_{n-1}$. Hence we have, $x_1 \leq x_2 \leq x_3 \leq ... \leq x_{n-4} \leq x_{n-3} \leq x_{n-2} < x_n < x_{n-1}$. Switch $x_n$ and $x_{n-1}$ in $\boldsymbol{x}$ and call the new sequence $\hat{\boldsymbol{x}}$.

$$\boldsymbol{x}=x_1, ..., x_{n-1}, x_n, x_{n+1}, ... \text{ and } \hat{\boldsymbol{x}}=x_1, ..., x_n, x_{n-1}, x_{n+1}, ....$$

Subtracting $E(\hat{\boldsymbol{x}})$ from $E(\boldsymbol{x})$ and canceling out terms yields:

$$E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) = f(x_{n-3})(x_{n-1} - x_n) + f(x_n)(x_{n+2} - x_{n-1}) + f(x_{n-1})(x_n - x_{n+2}).$$

Note that $f(x_{n-3}) > f(x_n) > f(x_{n-1})$ and $x_{n-1} - x_n > 0$. Also, by Theorem 2.2.3 $x_n - x_{n+2} < 0$. However, $x_{n+2} - x_{n-1}$ could be positive or negative. Suppose $x_{n+2} - x_{n-1} > 0$. Then by multiplying the positive terms $x_{n+2} - x_{n-1}$ and $x_{n-1} - x_n$ by a smaller number, $f(x_{n-1})$, yields:

$$E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) > f(x_{n-1})(x_{n-1} - x_n) + f(x_{n-1})(x_{n+2} - x_{n-1}) + f(x_{n-1})(x_n - x_{n+2}) = 0.$$

Suppose $x_{n+2} - x_{n-1} < 0$. Then by multiplying the negative terms $x_{n+2} - x_{n-1}$ and $x_n - x_{n+2}$ by a larger number, $f(x_{n-3})$ yields:

$$E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) > f(x_{n-3})(x_{n-1} - x_n) + f(x_{n-3})(x_{n+2} - x_{n-1}) + f(x_{n-3})(x_n - x_{n+2}) = 0.$$

Hence, $E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) > 0$. □

**Theorem 2.2.5.** *If $\boldsymbol{x}$ is a sequence with $x_n < x_{n-1}$, then there exists a modified sequence $\hat{\boldsymbol{x}}$ such that $\hat{\boldsymbol{x}}$ is monotone and $E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) \geq 0$ .*

*Proof.* If the hypothesis that $f(x)$ is strictly decreasing is removed from Theorem 2.2.4 then it can be shown that a search plan $\boldsymbol{x}$ with $x_n < x_{n-1}$ can be rearranged to be monotone without decreasing the cost. The

proof is the same expect the conclusion is $E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) \geq 0$ instead of $E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) > 0$. □

## 2.3 Existence of Optimal Search Plans

Now that we know that optimal search plans must be monotone, we are able to establish the necessary conditions for an optimal plan to to exist. First, we must formally define an optimal search plan. An optimal search, $\boldsymbol{x}^*$, is an search plan such that $E(\boldsymbol{x}^*) = E^*$ where

$$E^* = \inf_{\{x_k\}_{k=-\infty}^{\infty}} \left\{ \sum_{k=-\infty}^{\infty} x_k(f(x_{k-1}) + f(x_{k-2})) | x_k > 0, k \in \mathbb{N}, x_k \to c \right\}. \tag{2.5}$$

Notice in the definition of the expected minimum cost, bi-infinite sequences are considered. This is because, depending on properties of $f(x)$, it might be possible to always lower the cost by adding an initial excursion to the search plan.

**Proposition 2.3.1.** *If $f(x)$ is not Lipschitz at 0, then the cost can always be lowered by adding a term to the sequence.*

*Proof.* Let $\boldsymbol{x}$ be a search plan. Insert a new turning point, $y$, in the beginning of $\boldsymbol{x}$ and call the new sequence $\boldsymbol{y}$.

$$E(\boldsymbol{x}) = x_1 + x_2(f(x_1) + .5) + x_3(f(x_2) + f(x_1)) + ...$$

$$E(\boldsymbol{y}) = y + x_1(f(y) + .5) + x_2(f(x_1) + f(y)) + x_3(f(x_2) + f(x_1)) + ...$$

So, $E(\boldsymbol{x}) > E(\boldsymbol{y})$ if and only if

$$\frac{x_1 + x_2}{2} - y - f(y)(x_1 + x_2) > 0.$$

Which happens if and only if

$$x_1 + x_2 > \frac{y}{.5 - f(y)}.$$

Since $f(x)$ is not Lipschitz about 0, such a $y$ can always be chosen. □

Since bi-infinite sequences do not have an initial turning point, we do not consider them to be search plans. So, for an optimal search plan to exist $f(x)$ must be Lipschitz.

**Proposition 2.3.2.** *Suppose $f(x)$ is a Lipschitz function with Lipschitz constant, $C_L$. Let $\boldsymbol{x}$ be a monotone sequence with $x_{m+1} < \frac{1}{4C_L}$. Then the modified sequence, $\hat{\boldsymbol{x}}$, with all $x_j$ $j < m$ removed, will have a lower expected cost.*

*Proof.* Since $f(x)$ is decreasing and $x_{m+1} < \frac{1}{4C_L}$,

$$\frac{f(x_{m-1})}{x_{m+1}} \geq \frac{f(x_{m+1})}{x_{m+1}} = \frac{\frac{1}{2}}{x_{m+1}} + \frac{f(x_{m+1}) - \frac{1}{2}}{x_{m+1}} > 2C_L - C_L = C_L.$$

Since $f(0) = .5$ and $f(x)$ is Lipschitz, $\frac{\frac{1}{2} - f(x_{m-1})}{x_{m-1}} < C_L$. Thus, $\frac{\frac{1}{2} - f(x_{m-1})}{x_{m-1}} < \frac{f(x_{m-1})}{x_{m+1}}$. Rearranging terms yields:

$$\frac{x_{m+1}}{2} < x_{m-1}f(x_{m-1}) + x_{m+1}f(x_{m-1}).$$

Since $f(x)$ is decreasing and $\boldsymbol{x}$ is monotone,

$$\frac{x_{m+1}}{2} < x_{m-1}f(x_{m-2}) + x_{m+1}f(x_{m-1}). \tag{2.6}$$

Similarly,

$$\frac{x_m}{2} < x_{m-2}f(x_{m-3}) + x_m f(x_{m-2}). \tag{2.7}$$

Since $\frac{\frac{1}{2} - f(x_{m-1})}{x_{m-1}} < \frac{f(x_{m-2})}{x_m}$,

$$\frac{x_m}{2} < x_{m-1}f(x_{m-2}) + x_m f(x_{m-1}).$$

Since $f(x)$ is decreasing and $\boldsymbol{x}$ is monotone,

$$\frac{x_m}{2} < x_{m-1}f(x_{m-3}) + x_m f(x_{m-1}). \tag{2.8}$$

Also,

$$x_{m+1}f(x_m) < x_{m+1}f(x_m) + x_{m-2}f(x_{m-4}). \tag{2.9}$$

From inequalities (2.6), (2.7), (2.8) and (2.9) we get:

$$x_m + x_{m+1}(f(x_m) + .5) < \sum_{n=1}^{m+1} x_n(f(x_{n-1}) + f(x_{n-2}))$$

Adding $\sum_{n=m+2}^{\infty} x_n(f(x_{n-1}) + f(x_{n-2}))$ to both sides yields the desired result. □

Finally we need one last proposition before we can prove existence of an optimal search plan. If $f(x)$ is Lipschitz then a search plan cannot have a cluster point about 0 or any other point that is not $c$. Also, as we will see in the next proposition, $x_n$ cannot grow without bound.

**Proposition 2.3.3.** *Suppose $f(x)$ is Lipschitz. Let $c$ be as defined in (2.1). If $\boldsymbol{x}$ is a minimizing search plan there exist sequences $\boldsymbol{a}$ and $\boldsymbol{b}$ where $a_n \to c$, $b_n \to c$ and $a_n \leq x_n \leq b_n$. Note $c$ may or may not be finite.*

*Proof.* From the previous propositions we know $\boldsymbol{x}$ has no cluster point at 0. It can be checked that if there existed any other cluster point that is not $c$, the expected cost function would diverge to infinity. Hence, you can construct a sequence $\boldsymbol{a}$ such that $a_n \to c$ and $a_n \leq x_n$. If $c$ is finite, we can construct a sequence $\boldsymbol{b}$ such that $b_n < \infty$, $b_n \to c$ and $x_n \leq b_n$. If $c = \infty$, then for any minimizing sequence we must have: $x_1 \leq 4E^* = b_1$ and $x_2(.5 + f(x_1)) \leq 4E^*$. Hence,

$$x_2 \leq \frac{4E^*}{.5 + f(x_1)} \leq \frac{4E^*}{2f(x_1)} \leq \frac{2E^*}{f(b_1)} = b_2.$$

Similarly,

$$x_n(f(x_{n-2}) + f(x_{n-1})) \leq 4E^*.$$

So,

$$x_n \leq \frac{4E_2^*}{f(x_{n-2}) + f(x_{n-1})} \leq \frac{4E^*}{2f(x_{n-1})} = \frac{2E^*}{f(b_{n-1})} = b_n.$$

Since $xf(x) < L < E^* < 2E^*$, the mapping, $x = \dfrac{2E^*}{f(x)}$, has no fixed point. Thus $b_n$ is a sequence that grows monotonically and to infinity. □

Now that we know optimal sequences must be monotone and that they cannot have any cluster points around any point that is not $c$, we can prove existence. But first, we state Fatou's Lemma, as it will be used in the proof for existence.

**Theorem 2.3.4** (Fatou's Lemma)**.** *Suppose $X_n$ are random variables. If $X_n \geq 0$ then*

$$\mathbb{E}(\liminf(X_n)) \leq \liminf \mathbb{E}(X_n).$$

**Theorem 2.3.5.** *If $f(x)$ is Lipschitz then there exists a converging subsequence, $\boldsymbol{x^{n_m}} \to \boldsymbol{x^*}$, such that $\boldsymbol{x^*}$ is an optimal search plan.*

*Proof.* By the definition of infimum there exists a sequence, $\boldsymbol{x^n}$, such that $E(\boldsymbol{x^n}) \to E^*$. From Proposition 2.3.3, we know that there exists sequences $\{a_k\}_{k=0}^{\infty}$ and $\{b_k\}_{k=0}^{\infty}$ such that for $n$ large enough $a_k \leq x_k^n \leq b_k$. Since each $x_k^n$ is bounded we may chose a subsequence, $\boldsymbol{x^{n_m}}$, of $\boldsymbol{x^n}$ such that $x_k^{n_m} \to x_k^*$ as $n_m \to \infty$ for any $k$.

Now we need to show that $E(\boldsymbol{x}^*) = E^*$. To do this we want to show $D(\boldsymbol{l}, \boldsymbol{x}^{\boldsymbol{n_m}}) \to D(\boldsymbol{l}, \boldsymbol{x}^*)$. Note that the distance variable is deterministic if the location, $\boldsymbol{l}$, of the object is known. Since $D$ is a continuous function of finitely many variables, $D(\boldsymbol{l}, \boldsymbol{x}^{\boldsymbol{n_m}}) \to D(\boldsymbol{l}, \boldsymbol{x}^*)$. From Theorem 2.3.4 we have the inequality

$$\mathbb{E}(\liminf(D(\boldsymbol{l}, \boldsymbol{x}^{\boldsymbol{n_m}})) \leq \liminf \mathbb{E}(D(\boldsymbol{l}, \boldsymbol{x}^{\boldsymbol{n_m}})).$$

Since $D(\boldsymbol{l}, \boldsymbol{x}^{\boldsymbol{n_m}}) \to D(\boldsymbol{l}, \boldsymbol{x}^*)$ we have,

$$E^* \leq \mathbb{E}(D(\boldsymbol{l}, \boldsymbol{x}^*)) = \mathbb{E}(\liminf(D(\boldsymbol{l}, \boldsymbol{x}^{\boldsymbol{n_m}})) \leq \liminf \mathbb{E}(D(\boldsymbol{l}, \boldsymbol{x}^{\boldsymbol{n_m}})) = E^*.$$

Hence, $E(\boldsymbol{x}^*) = E^*$. □

Now that we have established existence of optimal sequences, we can show that the turning points of an optimal sequences must satisfy an underlying fourth order recurrence relation.

## 2.4 Recurrence Relation

Optimal search plans for the linear search on the real line and the blind search on one ray both have second order recursion relations [4, 14]. However, for the blind linear search on the real line, the recursion relation is fourth order.

**Proposition 2.4.1.** *An optimal sequence will satisfy:*

- $f(x_{n-1}) + f(x_{n-2}) + x_{n+1}f'(x_n) + x_{n+2}f'(x_n) = 0$
- $f''(x_n) \geq 0$

*Proof.* $E(\boldsymbol{x})$ depends on $x_n$ via $x_n(f(x_{n-1}) + f(x_{n-2})) + f(x_n)(x_{i+1} + x_{n+2})$. Differentiating $E(\boldsymbol{x})$ with respect to $x_n$ yields $f(x_{n-1}) + f(x_{n-2}) + f'(x_n)(x_{n+1} + x_{n+2}) = 0$. Differentiating $E(\boldsymbol{x})$ with respect to $x_n$ twice yields $f''(x_n) \geq 0$. □

At the beginning of the chapter, in Equation (2.3), it was mentioned that sometimes it is useful for $E(\boldsymbol{x})$ to be written as:

$$\sum_{n=-1}^{\infty} f(x_n)(x_{n+1} + x_{n+2}).$$

This is because if $\boldsymbol{x}$ is an optimal sequence, then $x_{n+1} + x_{n+2} = \dfrac{f(x_{n-1}) + f(x_{n-2})}{-f'(x_n)}$. So, for optimal

sequences we have $E(\boldsymbol{x})$ is equal to

$$f(x_{-1})(x_1+x_0)+f(x_0)(x_1+x_2)+\sum_{n=1}^{\infty} f(x_n)(x_{n+1}+x_{n+2}) = x_1+.5x_2+\sum_{n=1}^{\infty}\frac{-f(x_n)}{f'(x_n)}(f(x_{n-1})+f(x_{n-2})).$$

Now we give an example where this rearrangement of the cost function leads to something interesting.

**Example 2.4.1.** Suppose we have two, $(1,\infty)$, rays with $f(x)=\dfrac{1}{2x^2}$. Then the recurrence is given by:

$$x_{n+2}+x_{n+1} = x_n^3\left(\frac{1}{2x_{n-1}^2}+\frac{1}{2x_{n-2}^2}\right).$$

The cost function can be arranged as:

$$E(\boldsymbol{x}) = x_1+\frac{x_2}{2}+\sum_{n=1}^{\infty}\frac{x_n^3}{2x_n^2}\left(\frac{1}{2x_{n-1}^2}+\frac{1}{2x_{n-2}^2}\right) = x_1+\frac{x_2}{2}+\sum_{n=1}^{\infty}\frac{x_n}{2}\left(\frac{1}{2x_{n-1}^2}+\frac{1}{2x_{n-2}^2}\right) = x_1+\frac{x_2}{2}+\frac{E(\boldsymbol{x})}{2}.$$

So, $E(\boldsymbol{x}) = 2x_1+x_2$.

However, finding $x_1$ and $x_2$ that generate a monotone search plan which yields a finite expected cost is still difficult. Nevertheless, $E(\boldsymbol{x}) = 2x_1+x_2$ is still an interesting observation for an optimal search plan for $f(x)=\dfrac{1}{2x^2}$. Using Proposition 2.1.2, we can calculate that $E(\boldsymbol{x})\leq 9$. So, for optimal $x_1$ and $x_2$ we have $2x_1+x_2\leq 9$. In the analogous one ray case with $f(x)=\dfrac{1}{x^2}$ on $(1,\infty)$, it was proven analytically that $E(\boldsymbol{x})=2x_1=4$ [14].

# Chapter 3

# Alternating

In the original linear search problem optimal search plans are always alternating. This is because the searcher can detect the object when it comes across it. However, for the blind linear problem the searcher must return to the origin to verify if the object has been located. Since the searcher alternates between rays in the original linear search problem, for the blind linear search problem, one might assume that an optimal search plan on the real line would also be alternating. However, it is not necessarily clear that is the case. In the following example we will see a distribution with a bounded domain where both alternating and non-alternating search plans yield the minimum expected cost.

**Example 3.0.1.** Suppose we have the uniform distribution on $[-1, 1]$. Then we have $f(x) = .5 - .5x$. Since $f'(1) \neq 0$, from Proposition 2.1.3 we know an optimal search plan will have finitely many turning points. It can then be checked that an optimal sequence is given by $x_1 = 1$ and $x_2 = 1$. So, $E(\boldsymbol{x}) = 1+1(f(1)+f(0)) = 1+1(0+.5) = 1.5$. Now suppose we have a non-alternating search plan where the searcher went to 1 on the first ray then to $y$ on the second and then to 1 on the second ray.

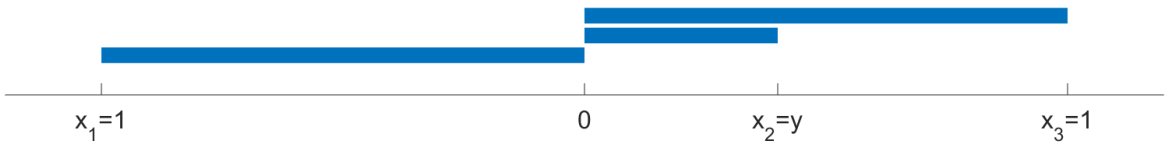


Figure 3.1: Non-alternating search plan that yields the expected minimum cost for $f(x) = .5(1 - x)$.

The expected cost for the non-alternating search plan would be $1 + y(f(1) + f(0)) + 1(f(1) + f(y)) =$

$1 + y(0 + .5) + 1(0 + .5 - .5y) = 1.5$. For $f(x) = .5 - .5x$, it can be checked that given any non-alternating sequence there is an alternating sequence that yields an equal or lesser expected cost. So, 1.5 is the minimum for both alternating and non-alternating search plans.

This example, albeit simple, leads to an interesting question: Is it always the case that there are alternating and non-alternating search plans that both yield the minimum cost? If not, how would one determine which type of sequence would generate the minimum? Or perhaps the uniform distribution is simple enough that this is only case where this happens. In this chapter we will investigate the conditions on $f(x)$ which guarantee an optimal search plan will be alternating between rays.

## 3.1 Notation and Terminology

When the search plan is not required to alternate between rays, keeping tracking of the the rays an excursion occurs on can be a cumbersome task. Thus, we introduce some new notation to keep track of the excursions, which rays they occur on and how they affect other excursions. When we say an excursion, $x_i$, "affects" another excursion, $x_j$, we mean $x_j f(x_i)$ occurs in the expected cost function. For example, when the search plan is alternating, $x_i$ affects $x_{i+1}$ and $x_{i+2}$. Also, in the alternating case $x_i$ is affected by $x_{i-1}$ and $x_{i-2}$. In both alternating and non-alternating search plans $x_i$ is always affected by $x_{i-1}$.

A search search plan, $\boldsymbol{x} = \{x_i, (s_i, a_i, b_i)\}_{i=1}^{\infty}$, in this section will be denoted by an infinite sequence of turning points where each turning point, $x_i$, has an accompanying triple, $(s_i, a_i, b_i)$.

- $s_i$ takes on values of 1 or $-1$. If $s_i = 1$, the excursion $x_i$ occurs on the positive ray. Similarly, If $s_i = -1$, the excursion $x_i$ occurs on the negative ray.
- $a_i$ takes on positive integer values and denotes how may excursions are affected by $x_i$.
- $b_i$ takes on positive integer values and denotes how far back in the sequence the previous excursion aside from $x_{i-1}$ which also affects $x_i$.

The following graphic gives an example of this notation for a search plan $\boldsymbol{x}$.

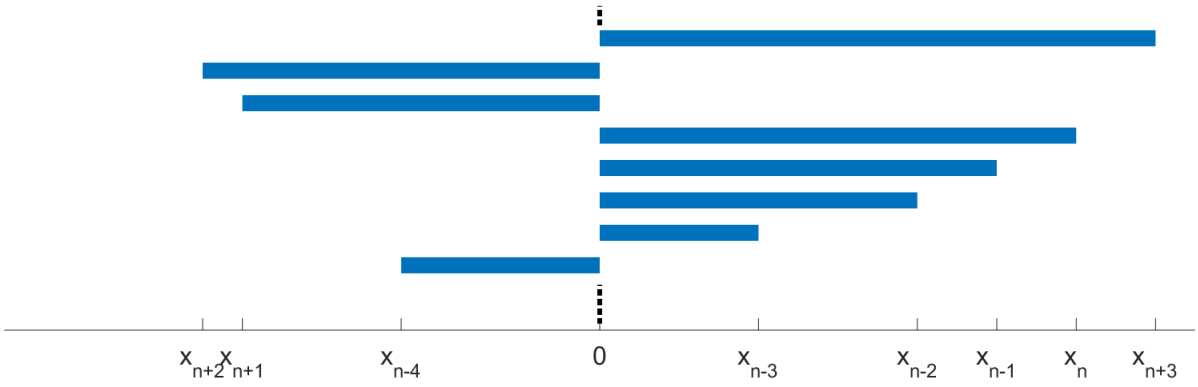


Figure 3.2: Visual representation of excursions from a non-alternating search plan, $\boldsymbol{x}$. $x_n$ is an excursion that occurs on the positive ray, affects $x_{n+1}, x_{n+2}$ and $x_{n+3}$ and is affected by $x_{n-1}$ and $x_{n-4}$. Hence, the triple, $(s_n, a_n, b_n)$, for $x_n$ would be $(1, 3, 4)$.

Similarly to Equation (2.3), the cost function for non-alternating search plans can be written as:

$$E(\boldsymbol{x}) = \sum_{n=-1}^{\infty} f(x_n)(x_{n+1} + ... + x_{n+a_n}). \tag{3.1}$$

Let $x_i$ be an excursion with triple $(s_i, a_i, b_i)$. Then $E(\boldsymbol{x})$ depends on $x_i$ via $x_i(f(x_{i-1}) + f(x_{i-b_i})) + f(x_i)(x_{i+1} + ... + x_{i+a_i})$. Analogously to Section 2.4, differentiating $E(\boldsymbol{x})$ with respect to $x_i$ yields the following recurrence relation:

- $f(x_{i-1}) + f(x_{i-b_i}) + f'(x_i)(x_{i+1} + ... + x_{i+a_i}) = 0$
- $f''(x_i) \geq 0$

Since there are infinitely many ways to alternate between rays, the condition $f(x_{i-1})+f(x_{i-b_i})+f'(x_i)(x_{i+1}+ ... + x_{i+a_i}) = 0$ probably is not very insightful or useful. However, the other condition, $f''(x_i) \geq 0$ is, as it will allow us to rule out points where $f''(x_i) < 0$.

## 3.2 Conditions for Alternating Search Plans

In this section we give the conditions on $f(x)$ which guarantee that optimal sequences will be alternating between rays.

**Theorem 3.2.1.** *If $\boldsymbol{x}$ is a search plan that begins with a monotone sequential sub-sequence of turning points, then alternating between the two rays for that sequential sub-sequence yields lower cost.*

*Proof.* Let $\boldsymbol{x}$ be a search plan. Let $\{x_{n-1}, x_n, x_{n+1}, ..., x_{m-1}\}$ be the first occurrence of a sequential sub-sequence of turning points that are not alternating between the two rays. So, $x_n$ is in that sub-sequence and is the first occurrence of a turning point in the sub-sequence not alternating between the rays. Let $\hat{\boldsymbol{x}}$ be a modified search plan such that $\hat{x}_i = x_i$ for all $i$, $\hat{s}_i = s_i$ for $i < n$ and $\hat{s}_i = -s_i$ for $i \geq n$. In other words $\hat{\boldsymbol{x}}$ is modified version of $\boldsymbol{x}$ such that all excursions starting at the $n$-th and onward occur on the opposite ray. The figures below provide visual representations of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$.

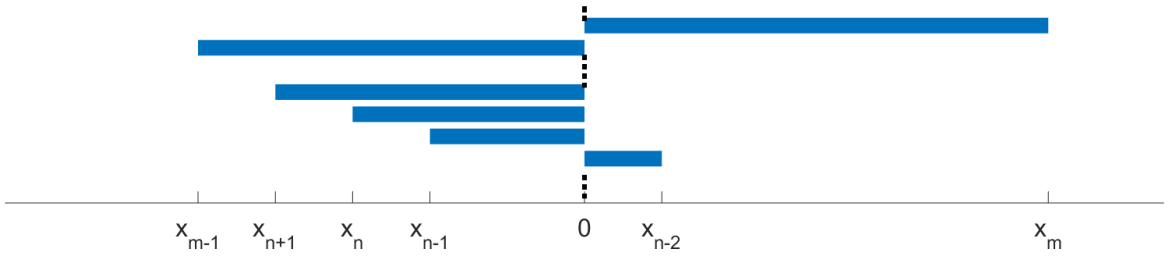


Figure 3.3: Excursions from $\boldsymbol{x}$.

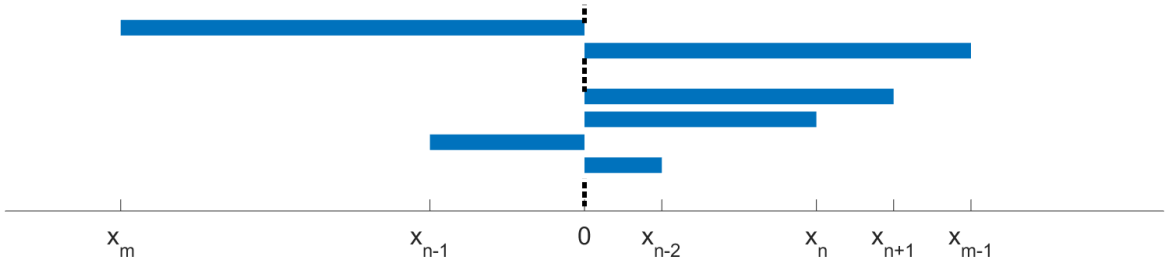


Figure 3.4: Excursions from $\hat{\boldsymbol{x}}$

In $E(\boldsymbol{x})$, the turning points $x_{n+1}, x_{n+2}, ..., x_m$ are affected by $x_{n-2}$. But, in $\hat{\boldsymbol{x}}$ the turning points $x_{n+1}, x_{n+2}, ..., x_m$ are now affected by $x_{n-1}$. Other than that, everything else remains the same. Hence, we have:

$$E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) = f(x_{n-2})(x_{n+1} + ... + x_m) - f(x_{n-1})(x_{n+1} + ... + x_m).$$

Since $x_{n-1} \geq x_{n-2}$ and $f(x)$ is decreasing,

$$E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) = f(x_{n-2})(x_{n+1} + ... + x_m) - f(x_{n-1})(x_{n+1} + ... + x_m) \geq 0.$$

□

**Corollary 3.2.2.** *An optimal sequence is either monotone and alternating or non-monotone and non-alternating.*

*Proof.* From Theorem 2.2.4, if a search plan is alternating and not monotone then it is not optimal. Similarly from Theorem 3.2.1, if a search plan is monotone and not alternating then it is not optimal. Thus, an optimal search plan is either monotone and alternating or non-monotone and non-alternating. □

In Example 3.0.1, we saw an $f(x)$ which had an alternating and non-alternating search plan both yield the minimum cost. This leads to the interesting and challenging question: Is there always at least one alternating and non-altering search plan that yields the minimum or does only one of them yield the minimum? If it turned out that only one of them yields the minimum, how would one determine which type of search plan yields the minimum? With so many different alternating patterns, if it turned out that non-alternating search plans yielded the minimum, it may be near impossible to investigate the underlying dynamics and solve the blind linear search problem. Thankfully, the following theorem gives conditions on a broad family of functions which guarantee optimal search plans must be alternating.

**Theorem 3.2.3.** *Suppose there exists a such that* $0 \leq a < \infty$. *If* $f''(x) < 0$ *for* $x \leq a$ *and* $f''(x) \geq 0$ *for* $x \geq a$, *then an optimal search plan will be alternating.*

*Proof.* Suppose $\boldsymbol{x}$ is an optimal search plan that is non-monotone and non-alternating. Since $\boldsymbol{x}$ is optimal, $f''(x_i) \geq 0$ for all $i$. Hence, $x_i \geq a$ for all $i$. Suppose $x_n$ is the first occurrence of non-monotonicity. Let $\hat{\boldsymbol{x}}$ be a modified search plan such that the searching order of $x_n$ and $x_{n-1}$ are swapped i.e., $\hat{x}_i = x_i$ for all $i$ except for $i = n-1$ and $i = n$. Let $\hat{x}_{n-1} = x_n$ and $\hat{x}_n = x_{n-1}$. The figures below provide visual representations of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$.

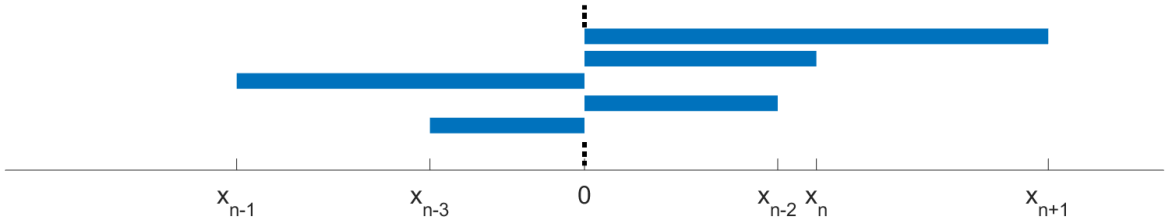


Figure 3.5: Excursions from $\boldsymbol{x}$.

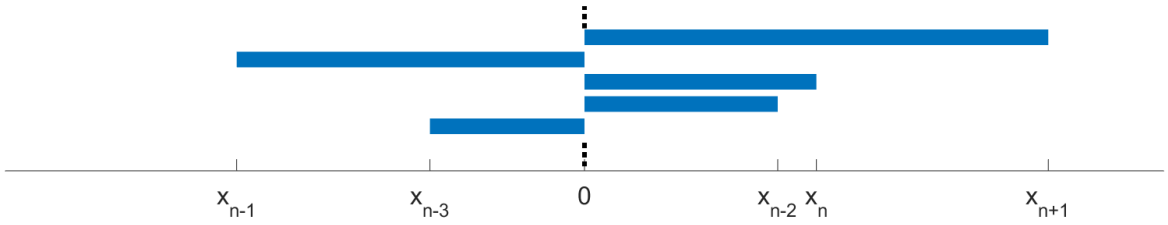


Figure 3.6: Excursions from $\hat{\boldsymbol{x}}$

The expected cost functions for both search plans only differ in terms with $x_n$ or $x_{n-1}$.

$$E(\boldsymbol{x}) = x_1 + ... + x_{n-1}(f(x_{n-2}) + f(x_{n-3})) + x_n(f(x_{n-1}) + f(x_{n-2})) + ...$$

$$E(\hat{\boldsymbol{x}}) = x_1 + ... + x_n(f(x_{n-2}) + f(x_{n-3})) + x_{n-1}(f(x_n) + f(x_{n-3})) + ...$$

Subtracting $E(\hat{\boldsymbol{x}})$ from $E(\boldsymbol{x})$ yields:

$$E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) = x_n(f(x_{n-1}) - f(x_{n-3})) + x_{n-1}(f(x_{n-2}) - f(x_n))$$
$$= x_n(f(x_{n-1}) - f(x_{n-3})) + x_{n-1}(f(x_{n-2}) - f(x_n)) + (x_{n-2} - x_{n-2})(f(x_{n-2}) - f(x_n)).$$

So, $E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) \geq 0$ if,

$$x_n(f(x_{n-1}) - f(x_{n-3})) + x_{n-1}(f(x_{n-2}) - f(x_n)) + (x_{n-2} - x_{n-2})(f(x_{n-2}) - f(x_n)) \geq 0.$$

So, $E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) \geq 0$ if,

$$(x_{n-1} - x_{n-2})(f(x_{n-2}) - f(x_n)) \geq x_n(f(x_{n-3}) - f(x_{n-1})) - x_{n-2}(f(x_{n-2}) - f(x_n)).$$

Since $x_{n-3} \leq x_{n-2} \leq x_n \leq x_{n-1}$ and $f(x)$ is decreasing,

$$x_n(f(x_{n-3}) - f(x_{n-1})) - x_{n-2}(f(x_{n-2}) - f(x_n)) \geq x_n(f(x_{n-2}) - f(x_{n-1})) - x_{n-2}(f(x_{n-2}) - f(x_{n-1})).$$

So, $E(\boldsymbol{x}) - E(\hat{\boldsymbol{x}}) \geq 0$ if,

$$(x_{n-1} - x_{n-2})(f(x_{n-2}) - f(x_n)) \geq (x_n - x_{n-2})(f(x_{n-2}) - f(x_{n-1})).$$

The inequality above holds if,

$$\frac{f(x_{n-2}) - f(x_n)}{x_n - x_{n-2}} \geq \frac{f(x_{n-2}) - f(x_{n-1})}{x_{n-1} - x_{n-2}}.$$

Since $x_i \geq a$ for all $i$ and $f(x)$ is convex for $x \geq a$, the above inequality holds. Thus, $E(\boldsymbol{x}) \geq E(\hat{\boldsymbol{x}})$. However, $\hat{\boldsymbol{x}}$ is not alternating at $x_n$ but $\hat{\boldsymbol{x}}$ is monotone up to $x_n$. Hence, using Theorem 3.2.1, an alternating search plan can be found by modifying $\hat{\boldsymbol{x}}$ that yields a lower expected cost than $\hat{\boldsymbol{x}}$. □

Recall that $p(x)$ is symmetric about the origin and recall the relation between $f(x)$ and $p(x)$ (Equation 2.4). So, $f''(x) \geq 0$ for all $x \geq 0$ corresponds to unimodal probability distributions. While, $f''(x) < 0$ for $0 \leq x < a$ and $f''(x) \geq 0$ for $x \geq a$, where $a$ is finite, corresponds to bimodal probability distributions. The following graphic gives a visual representation of: $f''(x) < 0$ for $0 \leq x < a$ and $f''(x) \geq 0$ for $x \geq a$.

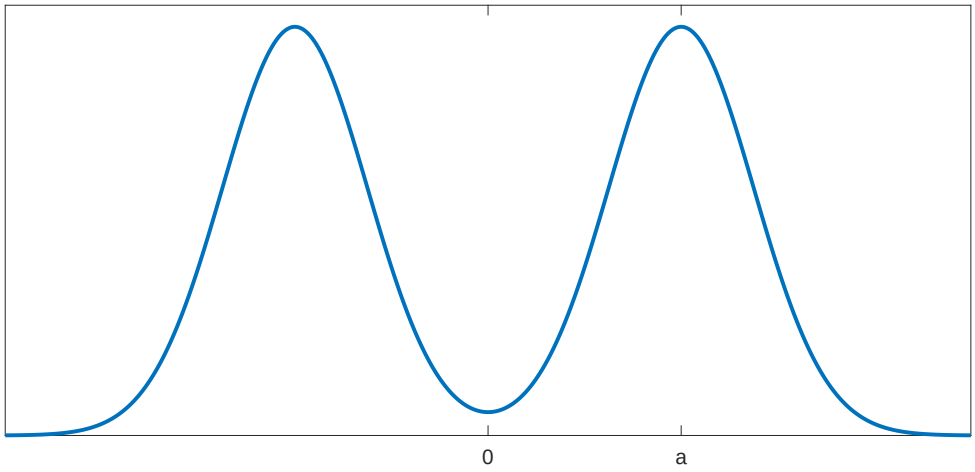


Figure 3.7: A graph of a bimodal distribution, $p(x)$. For $x \geq 0$, $f''(x) = -p'(x)$. On $(0, a)$, $f''(x) < 0$ and so $p(x)$ is increasing. On $(a, \infty)$, $f''(x) \geq 0$ and so $p(x)$ is decreasing.

Hence, Theorem 3.2.3 states when $p(x)$ is a unimodal or bimodal distribution, optimal search plans will always be alternating. Notice in Example 3.0.1, $f''(x) = 0$ for all $x$, so, Theorem 3.2.3 does not apply.

# Chapter 4

# Dynamic Program

In this section we present a dynamic programming algorithm which can be used to produce a search plan that approximates the minimum expected cost for the blind linear search on two rays up to any accuracy. To paraphrase the book, *Introduction to Algorithms* [8]: "a dynamic programming solves problems by combining the solutions to sub-problems. Dynamic programming applies when sub-problems share sub-sub-problems. A dynamic-programming algorithm solves each sub-sub-problem just once and then saves its answer in a table, thereby avoiding the work of recomputing the answer every time it solves each sub-sub-problem."

Robertson and Bruss first presented a dynamic programming algorithm that solves the original linear search problem with a discrete probability function on a finite set of points [7]. Washburn used a variation of this dynamic program to find the best starting point for the linear search problem [12]. The starting point refers to the starting position of the searcher which is not restricted to be the origin. Alpern and Gal then extended the dynamic program to approximate the linear search problem with any continuous distribution to any accuracy [1]. The algorithm presented in this section can solve the blind linear search problem with a discrete distribution on a finite set of points. We first show how the algorithm can be used to estimate the blind linear search with a continuous probability distribution on a finite domain. We then show how it can be extended to approximate the blind linear search problem with a continuous distribution on an infinite domain. Next, we give examples of the dynamic program and compare the results to results from Chapter 5. In the examples, the estimated expected minimum costs are close in value to results from Chapter 5. Also, in the examples, the estimated initial turning points, $x_1$ and $x_2$, are close in value to results from Chapter 5. This is interesting due to the fact the dynamic program does not take into account the underlying recurrence relation but still generates $x_1$ and $x_2$ that are close to the actual optimal $x_1$ and $x_2$. In the appendix we give an algorithm for the blind linear search problem on one ray.

## 4.1 Algorithm

In this section we present a dynamic program which can be used to solve the blind linear search problem with a discrete distribution on a finite set of points. For unimodal and bimodal distributions, optimal search plans must be alternating. So, the algorithm only considers alternating search plans.

After presenting the algorithm, we show how it can be extended to approximate the blind linear search problem with a continuous probability distribution. Before we present the dynamic program algorithm, we make note of a scale invariance property of $E(\boldsymbol{x})$. The scale invariance property will allow for easier computations when programming the algorithm.

**Remark 4.1.1.** *The expected cost function, $E(\boldsymbol{x})$, has a scale invariance property.*

Let $\boldsymbol{x}$ and $\boldsymbol{x}_\alpha = \alpha\boldsymbol{x}$ be search plans for $f(x)$ and $f_\alpha(x) = f(x/\alpha)$ respectively. Let $E(\boldsymbol{x})$ and $E_\alpha(\boldsymbol{x}_\alpha)$ be the respective associated expected costs. Then it can be readily checked that $E_\alpha(\boldsymbol{x}_\alpha) = \alpha E(\boldsymbol{x})$.

We now present the algorithm. Let $\boldsymbol{l}$ be the associated random variable for the discrete probability mass function, $p(x)$. Let $f(x) = P(\boldsymbol{l} > |x|)$ and let $E^*$ denote the minimum expected cost. Since the cost function has a scale invariance property (See Remark 4.1.1), we may assume the domain is $[-n, n]$ and $p(x)$ takes on values when $x$ is an integer. After using the dynamic program, the resulting minimum cost and turning points can be scaled appropriately.

If the searcher is out on an excursion at point $k$, the minimum expected remaining cost at point $k$ is the additional expected cost to get to the next point plus the minimum expected remaining cost at that next point. Let $L(i, j, k)$ denote the minimum expected remaining cost given that the object is not in the interval $[i, j]$ and the searcher currently on an excursion at point $k$ on the left ray. Similarly, let $R(i, j, k)$ denote the minimum expected remaining cost given that the object is not in the interval $[i, j]$ and the searcher is currently on an excursion at point $k$ on the right ray.

We begin by defining the expected minimum remaining costs at the end points. After the searcher visits $n$ on each ray, the search will be over. So, we have:

$$L(i, n, n) = 0$$

$$R(n, j, n) = 0$$

If $k = n$, $i < n$ and $j < n$, then the searcher will have had covered the whole ray on its next return to the origin. Since optimal search plans are monotone, the next excursion would be going to point $n$ on the other ray (See Figures 4.1 and 4.2). Hence, $n$ times the probability of the object not being located yet is

the additional expected cost to get to point $n$ on the other ray. So, we have:

$$L(i, j, n) = n(f(n) + f(j)) + R(n, j, n) = nf(j)$$

$$R(i, j, n) = n(f(i) + f(n)) + L(i, n, n) = nf(i)$$

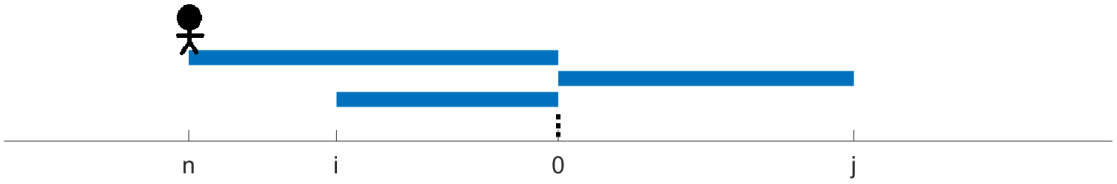


Figure 4.1: $L(i, j, n)$

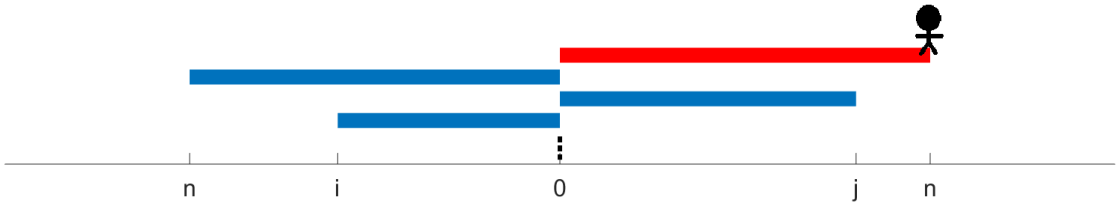


Figure 4.2: $R(n, j, n)$. If the searcher is at $L(i, j, n)$, then the searcher's next excursion must be to $R(n, j, n)$. The expected additional cost of this move is $n(f(n) + f(j)) = nf(j)$.

Since optimal search plans are monotone, $k \geq i$ and $k \geq j$. We next define the minimum expected cost when $k = j$ or $k = i$. For $L(i, j, k)$, if $j = k$, the searcher has covered $[i, j]$ and is currently at the point $j$ on the left ray. At $L(i, j, k)$, the searcher has two options:

- Begin a new excursion to the point $j + 1$ on the right ray.
- Continue its current excursion on the left ray by going to $j + 1$ on the left ray.

If the searcher chooses to begin a new excursion to $j + 1$ on the right ray, the additional distance traveled from point $j$ on the left ray to point $j + 1$ to the right ray would be $j + 1$. The additional expected cost of this move would be $(j + 1)$ times $f(j) + f(j)$. If the searcher chooses to continue its current excursion on the left ray by moving to $j + 1$ on the left ray, the additional distance traveled from point $j$ on the left ray to point $j + 1$ to the left ray would be 1. The additional expected cost of this move would be 1 times

$f(i) + f(j)$. So, if $j = k$:

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

Similarly, if $i = k$:

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

Otherwise, $k > i$ and $k > j$. Then for $L(i, j, k)$, the searcher has covered $[i, j]$ and is currently at the point $k$ on the left ray (See Figure 4.3). At $L(i, j, k)$, the searcher has two options:

- Begin a new excursion to the point $k$ on the right ray.
- Continue its current excursion on the left ray by going to $k + 1$ on the left ray.

If the searcher chooses to begin a new excursion to $k$ on the right ray the additional distance traveled from point $k$ on the left ray to point $k$ on the right ray would be $k$. The additional expected cost of this move would be $k$ times $f(k) + f(j)$. If the searcher chooses to continue its current excursion by going to $k + 1$ on the left ray, the additional distance traveled from point $k$ on the left ray to point $k + 1$ on the left ray would 1. The additional expected cost of this move would be 1 times $f(i) + f(j)$. (See Figures 4.3, 4.4 and 4.5). Thus we have:

$$L(i, j, k) = \min\{1(f(i) + f(j)) + L(i, j, k + 1), k(f(k) + f(j)) + R(k, j, k)\}$$

$$R(i, j, k) = \min\{k(f(i) + f(k)) + L(i, k, k), 1(f(i) + f(j)) + R(i, j, k + 1)\}$$

Lastly, if the searcher is at the origin and has not searched any interval, its only choice would be to move right or left one. Hence, we have:

$$E^* = L(0, 0, 0) = R(0, 0, 0) = 1 + R(0, 0, 1) = 1 + L(0, 0, 1).$$

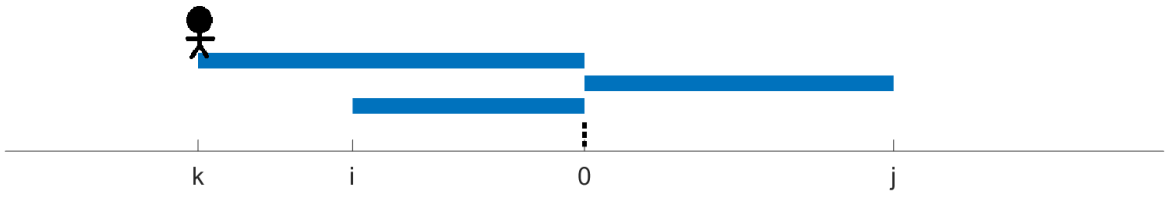


Figure 4.3: $L(i, j, k)$

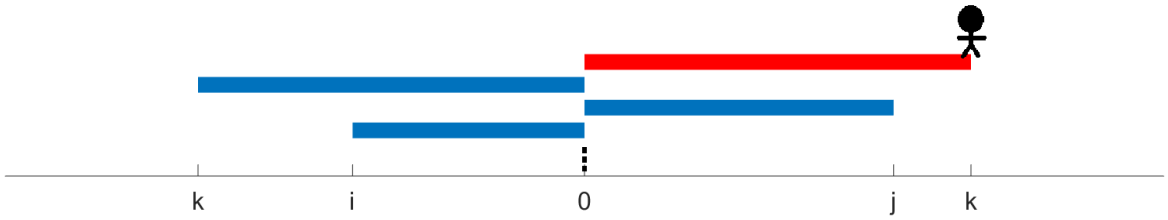


Figure 4.4: The searcher begins a new excursion by going from $k$ on the left ray to $k$ on the right ray. It takes the searcher an expected additional $k(f(k) + f(j))$ to go from $L(i, j, k)$ to $R(k, j, k)$.

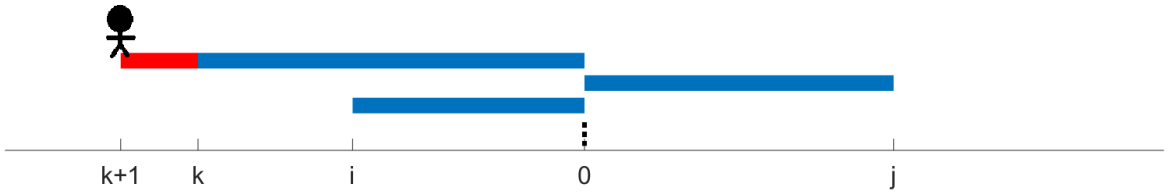


Figure 4.5: The searcher continues its current excursion on the left ray by going from $k$ on the left ray to $k + 1$ on the left ray. It takes the searcher an expected additional $1(f(i) + f(j))$ to go from $L(i, j, k)$ to $L(i, j, k + 1)$.

Having to check all possible search plans would take on the order of $O(2^n)$ computations. For the algorithm above, since all the computations can be stored in two $n \times n \times n$ arrays, the algorithm is of $O(n^3)$ complexity. Note the algorithm calculated sub-problems that shared other sub-problems. Hence, why it is a dynamic program.

**Proposition 4.1.2.** *Let $f(x)$ be a continuous distribution with finite domain. Let $E^*$ be the corresponding*

*expected minimum cost for the blind linear search problem with $f(x)$. Let $\alpha > 0$. Suppose $f_T(x)$ is a discretization of $f(x)$. Let $E_T^*$ be the corresponding expected minimum cost for $f_T(x)$. If $f_T(x)$ is chosen appropriately, then $|E_T^* - E^*| < \alpha$.*

*Proof.* Suppose we have a continuous symmetric probability density function, $p(x)$, with finite domain, $[-c, c]$. Let $\epsilon > 0$ and $\alpha > 0$. Partition the real line into sub-intervals of length $\epsilon$. Call this partition an $\epsilon$-grid. Using this $\epsilon$-grid we then discretize the probability. Let $p_d(x)$ be the discretization of $p(x)$. To generate $p_d(x)$, for $i \geq 0$, all of the probability mass of $(i\epsilon, (i+1)\epsilon)$ gets placed on the point $(i+1)\epsilon$ and the all of the probability mass $(-(i+1)\epsilon, -i\epsilon)$ gets placed on the point $-(i+1)\epsilon$. However, it could be the case that for some $N$, $(N-1)\epsilon \leq c < N\epsilon$. Generate a new probability mass function, $p_T(x)$, by truncating $p_d(x)$. To generate $p_T(x)$ take $p_d(x)$ but place all of the probability mass from $(-c, -(N-1)\epsilon)$ onto point $-c$ and all of the probability mass from $((N-1)\epsilon, c)$ onto point $c$. Let $\boldsymbol{l}_T$ be the associated random variable for $p_T(x)$. Let $\boldsymbol{l}_{\epsilon^+} = \boldsymbol{l} + \text{sign}(\boldsymbol{l})\epsilon$ be the associated random variable for the probability density function,

$$p_{\epsilon^+}(x) = \begin{cases} p(x+\epsilon) & x < -\epsilon \\ 0 & -\epsilon \leq x \leq \epsilon \\ p(x-\epsilon) & x > \epsilon \end{cases}$$

Notice that as $\epsilon \to 0$, $p_{\epsilon^+}(x) \to p(x)$. Define $f(x) = P(\boldsymbol{l} > |x|)$, $f_T(x) = P(\boldsymbol{l}_T > |x|)$ and $f_{\epsilon^+}(x) = P(\boldsymbol{l}_{\epsilon^+} > |x|)$. Let $E^*, E_T^*$ and $E_{\epsilon^+}^*$ denote the minimum expected costs for the random variables $\boldsymbol{l}$, $\boldsymbol{l}_T$ and $\boldsymbol{l}_{\epsilon^+}$ respectively. From how $p(x)$, $p_T(x)$ and $p_{\epsilon^+}(x)$ are defined, we have $f(x) \leq f_T(x) \leq f_{\epsilon^+}(x)$. Thus, any search plan for $\boldsymbol{l}_{\epsilon^+}$ yields a lower expected cost for $\boldsymbol{l}_T$ and any search plan for $\boldsymbol{l}_T$ yields a lower expected cost for $\boldsymbol{l}$. Hence, we have $E^* \leq E_T^* \leq E_{\epsilon^+}^*$.

Suppose $\boldsymbol{x}$ is a search plan with monotone turning points for $\boldsymbol{l}$. Then $\boldsymbol{x} + \epsilon = \{x_i + \epsilon\}_{i=1}^{\infty}$, is a search plan for $\boldsymbol{l}_{\epsilon^+}$. It can be checked by plugging $\{x_i + \epsilon\}_{i=1}^{\infty}$ into $E_{\epsilon^+}(\boldsymbol{x}) = \sum_{n=1}^{\infty} x_n(f_{\epsilon^+}(x_{n-1}) + f_{\epsilon^+}(x_{n-2})) = \sum_{n=1}^{\infty} x_n(f(x_{n-1}-\epsilon) + f(x_{n-2}-\epsilon))$ yields $E_{\epsilon^+}(\boldsymbol{x}+\epsilon) = \sum_{n=1}^{\infty}(x_n+\epsilon)(f(x_{n-1}) + f(x_{n-2})) = \sum_{n=1}^{\infty}(x_n)(f(x_{n-1}) + f(x_{n-2})) + \epsilon\sum_{n=1}^{\infty}(f(x_{n-1}) + f(x_{n-2}))$. Let $L = \int_{-c}^{c} f(x)\,dx = \int_0^c 2f(x)\,dx$. Since $f(x)$ is a decreasing function, $\boldsymbol{x}$ can be chosen such that:

$$\epsilon\sum_{n=1}^{\infty}(f(x_{n-1}) + f(x_{n-2})) \leq \epsilon L.$$

Hence, $E^* \leq E_{\epsilon^+}^* \leq E^* + \epsilon L$. Thus we have: $E^* \leq E_T^* \leq E_{\epsilon^+}^* \leq E^* + \epsilon L \leq E_T^* + \epsilon L$ or, equivalently, $E_T^* - \epsilon L \leq E^* \leq E_T^*$.

For any epsilon we can use the dynamic programming algorithm to generate a search plan, $\boldsymbol{x}_T^*$, that yields $E_T^*$. From how $f(x)$ and $f_T(x)$ are defined, $f(x) = f_T(x)$ for any $x$ in the set $\{0, \epsilon, 2\epsilon, 3\epsilon, ..., (N-1)\epsilon, c\}$. So, plugging $\boldsymbol{x}_T^*$ into $E(\boldsymbol{x})$ yields $E(\boldsymbol{x}_T^*) = E_T^*$. Finally we have the inequality $E(\boldsymbol{x}_T^*) - \epsilon L \leq E^* \leq E(\boldsymbol{x}_T^*)$. Letting $\epsilon = \dfrac{\alpha}{L}$ yields the desired result of $|E_T^* - E^*| < \alpha$. □

Next we show how the algorithm can approximate the blind linear search problem with any distribution on an infinite domain.

**Proposition 4.1.3.** *Let $f(x)$ be a continuous distribution with infinite domain. Let $E^*$ be the corresponding expected minimum cost for the blind linear search problem with $f(x)$. Let $\alpha > 0$. Suppose $f_T(x)$ is a discretization of $f(x)$. Let $E_T^*$ be the corresponding expected minimum cost for $f_T(x)$. If $f_T(x)$ is chosen appropriately, then $|E_T^* - E^*| < 2\alpha$.*

*Proof.* Suppose $p(x)$ is a continuous symmetric probability distribution function on $(-\infty, \infty)$. Let $\epsilon > 0$, $\alpha > 0$ and $f(x) = P(\boldsymbol{l} > |x|)$. Choose $C$ such that:

$$Cf(C) + 2C(f(C) + f(C)) + \sum_{i=0}^{\infty} 2^{i+2} C(f(2^{i+1}C) + (f(2^i C)) \leq \epsilon L \tag{4.1}$$

Using a similar argument as the one used in the proof of Proposition 2.1.1, such a C can be chosen. Later we will see why such a $C$ was chosen. The distributions $p_d(x)$ and $p_{\epsilon^+}(x)$ are generated the same way as in the previous proposition. Define $f(x) = P(\boldsymbol{l} > |x|)$, $f_d(x) = P(\boldsymbol{l}_d > |x|)$ and $f_{\epsilon^+}(x) = P(\boldsymbol{l}_{\epsilon^+} > |x|)$. Let $E^*, E_d^*$ and $E_{\epsilon^+}^*$ denote the minimum expected costs for the random variables $\boldsymbol{l}$, $\boldsymbol{l}_d$ and $\boldsymbol{l}_{\epsilon^+}$ respectively. From how $p(x)$, $p_d(x)$ and $p_{\epsilon^+}(x)$ are defined, we have $f(x) \leq f_d(x) \leq f_{\epsilon^+}(x)$. Thus, any search plan for $\boldsymbol{l}_\epsilon^+$ yields a lower expected cost for $\boldsymbol{l}_d$ and any search plan for $\boldsymbol{l}_d$ yields a lower expected cost for $\boldsymbol{l}$. Hence, we have $E^* \leq E_d^* \leq E_{\epsilon^+}^*$. Suppose $\boldsymbol{x}$ is a search plan with monotone turning points for $\boldsymbol{l}$. Then $\boldsymbol{x} + \epsilon = \{x_i + \epsilon\}_{i=1}^{\infty}$, is a search plan for $\boldsymbol{l}_{\epsilon^+}$. It can be checked by plugging $\{x_i + \epsilon\}_{i=1}^{\infty}$ into $E_{\epsilon^+}(\boldsymbol{x}) = \sum_{n=1}^{\infty} x_n(f_{\epsilon^+}(x_{n-1}) + f_{\epsilon^+}(x_{n-2})) = \sum_{n=1}^{\infty} x_n(f(x_{n-1} - \epsilon) + f(x_{n-2} - \epsilon))$ yields $E_{\epsilon^+}(\boldsymbol{x} + \epsilon) = \sum_{n=1}^{\infty}(x_n + \epsilon)(f(x_{n-1}) + f(x_{n-2})) = \sum_{n=1}^{\infty}(x_n)(f(x_{n-1}) + f(x_{n-2})) + \epsilon \sum_{n=1}^{\infty}(f(x_{n-1}) + f(x_{n-2}))$. Let $L = \int_{-\infty}^{\infty} f(x)\,dx = \int_0^{\infty} 2f(x)\,dx$. Since $f(x)$ is a decreasing function, $\boldsymbol{x}$ can be chosen such that:

$$\epsilon \sum_{n=1}^{\infty}(f(x_{n-1}) + f(x_{n-2})) \leq \epsilon L.$$

Hence, $E^* \leq E_{\epsilon^+}^* \leq E^* + \epsilon L$.

The distributions $p(x)$, $p_d(x)$ and $p_{\epsilon^+}(x)$ all have infinite domains. However, the algorithm only solves

for the minimum expected cost for a discrete probability with a finite set of points. Hence, we need to define a new probability mass function, $p_T(x)$, that is created by truncating $p_d(x)$. To generate $p_T(x)$ take $p_d(x)$ and place all of the probability mass from $(-\infty, -C)$ onto point $-C$ and all of the probability mass from $(C, \infty)$ onto point $C$. Let $\boldsymbol{l}_T$ be the associated random for $p_T(x)$.

Let $\boldsymbol{x}_T^* = n_1\epsilon, n_2\epsilon, ..., n_{N-1}\epsilon, C, C$, where $n_i\epsilon$ is a point on the $\epsilon$-grid less than C, be the sequence for $\boldsymbol{l}_T$ that yields minimum expected cost $E_T^*$. We can create a search plan from $\boldsymbol{x}_T^*$ that extends to infinite domains. Let $f_T(x) = P(\boldsymbol{l}_T > |x|)$. Let $\boldsymbol{x}_\infty = n_1\epsilon, n_2\epsilon, ..., n_{N-1}\epsilon, C, C, 2C, 4C, 8C, 16C, ....$ Now we want to look at the difference of $E(\boldsymbol{x}_\infty)$ and $E_T(\boldsymbol{x}_T^*) = E_T^*$.

If $x$ is a point on the $\epsilon$-mesh less than C, then $f(x) = f_T(x)$. Note $f_T(x) = 0$ for $x \geq C$. Subtracting $E_T(\boldsymbol{x}_T^*)$ from $E(\boldsymbol{x}_\infty)$ and canceling out terms yields,

$$E(\boldsymbol{x}_\infty) - E_T(\boldsymbol{x}_T^*) = Cf(C) + 2C(f(C) + f(C)) + \sum_{i=0}^{\infty} 2^{i+2}C(f(2^{i+1}C) + (f(2^iC)) \leq \epsilon L.$$

From how $p_d(x)$ and $p_T(x)$ are defined we have $f_T(x) \leq f_d(x)$. Hence any search plan for $\boldsymbol{l}_d$ yields a lower search plan for $\boldsymbol{l}_T$. Hence, $E_T^* \leq E_d^*$. Finally we have: $E_T^* \leq E_d^* \leq E_{\epsilon^+} \leq E^* + \epsilon L \leq E(\boldsymbol{x}_\infty) + \epsilon L \leq E_T^* + \epsilon 2L$. So, $E_T^* - \epsilon L \leq E^* \leq E(\boldsymbol{x}_\infty) \leq E_T^* + \epsilon L$. Letting $\epsilon = \dfrac{\alpha}{L}$ yields the desired result of $|E_T^* - E^*| < 2\alpha$. □

The number of calculations required for a desired accuracy depends on the probability function and the $\epsilon$-grid. In the next section we use the algorithm to estimate the minimum expected cost for the blind linear search problem for two different distributions. In the examples, we show how to calculate the accuracy of the estimates and the number of computations required.

## 4.2 Examples

**Example 4.2.1.** The dynamic program with $f(x) = .5\exp(-x)$ on the interval $[-9, 9]$ with a .05-grid yields $x_1 = .5$, $x_2 = 1.2$ and an expected minimum cost of 3.6848. Since $9/.05 = 180$, $n = 180$. It can be checked that $C = 9$ satisfies (4.1). So, this estimate took on the order of $180^3$ computation and guarantees an accuracy up to $10^{-1}$. The initial points from Chapter 5 are $x_1 \approx .4931...$ and $x_2 \approx 1.1783...$ with an approximated expected cost of 3.6826.

**Example 4.2.2.** The dynamic program with $f(x) = .5\exp(-x^2)$ on the interval $[-6, 6]$ with a .04-grid yields $x_1 = 1.2$, $x_2 = 1.56$ and an expected minimum cost of 2.6524. Since $6/.04 = 150$, $n = 150$. It can be checked that $C = 6$ satisfies (4.1). So, this estimate took on the order of $150^3$ computations and guarantees an accuracy up to .08. The initial points from Chapter 5 are $x_1 \approx 1.223...$ and $x_2 \approx 1.565...$ with

an approximated expected cost of 2.6585.

The initial turning points calculated from the dynamic program are close in value to the initial turning points from Chapter 5. In the appendix, examples from the dynamic program for the one ray case are also given. Since the algorithm for one ray case requires less computing power, a finer grid was used. So for the one ray case, the initial turning points from the dynamic program and invariant curves are closer in value. If one had more computing power, it would be reasonable to assume that using a finer grid on the two ray case would generate more accurate initial turning points and minimum expected cost.

# Chapter 5

# Searching for Optimal Initial Tuning Points

The objective of this chapter is to numerically find invariant curves for (5.1) on which optimal initial turning points lay. For $f(x) = .5\exp(-x)$, we use numerical evidence to formulate a conjecture that optimal initial turning points begin on a one dimensional invariant curve in the phase space. In the analogous one ray case, Baryshnikov and Zharnitsky analytically proved that there existed separatrix which was an invariant curve in the phase space that contained optimal turning points [14]. Since the phase space in the one ray case is of two dimensions, a curve can be a candidate for a separatrix. However, for the two ray case the phase space is of four dimensions. So, it is plausible that no separatrix exists. However, using what is known about the one ray case, we make careful estimates and show there is numerical evidence for a one dimensional invariant curve in the phase space of (5.1) where optimal turning points lay. Then at the end of the chapter, we give a procedure that aids in numerically finding one dimensional invariant curves where optimal initial turning points lay.

## 5.1 Laplace Distribution

In the one ray case with $f(x) = \exp(-x)$, iterations on the boundary of monotonicity converge to the separatrix. When iterations from the inverse map (1.2) start on the boundary of monotonicity, $x_n$ and $y_n$ are always defined [14]. This is because $x_n$ is always positive. So, $y_n = \ln(x_{n+1})$ is always defined. However, for the two ray case it is not clear if it is possible to analytically prove iterations on the boundary of monotonicity are always defined and convergent.

For the analogous two rays case, we have $f(x) = .5\exp(-x)$. The recursion relation is given by:

$$x_{n+1} + x_{n+2} = \exp(x_n - x_{n-1}) + \exp(x_n - x_{n-2})$$

Using a change of variables yields:

$$\begin{aligned} x_{n+1} &= u_n - x_n \\ u_{n+1} &= \exp(y_n) + \exp(z_n) \\ y_{n+1} &= x_{n+1} - x_n \\ z_{n+1} &= y_{n+1} + y_n \end{aligned} \tag{5.1}$$

The inverse map is given by:

$$\begin{aligned} x_n &= x_{n+1} - y_{n+1} \\ u_n &= x_n + x_{n+1} \\ y_n &= z_{n+1} - y_{n+1} \\ z_n &= \log(u_{n+1} - \exp(y_n)) \end{aligned} \tag{5.2}$$

In the one ray case, backwards iterations converged to the separatrix which was an invariant curve. So, for the to ray case, we are interested in the convergence of backwards iterations. When iterating the inverse map (5.2), it is not clear if $z_n$ will always be defined. For $z_n$ to always be defined, this would require $u_{n+1} - \exp(y_n)$ to be positive at each iteration. However, it not easy to guarantee $u_{n+1} - \exp(y_n)$ will always be positive at each iteration. Nonetheless, as we will see in the next subsection, we are able to find a curve which can be iterated backwards and have $z_n$ be defined at each iteration.

Optimal sequences are defined by the initial choice of $x_1$ and $x_2$. Recall, $x_{-1}$ and $x_0$ are taken to be zero. So, we have $x_2 + x_3 = \exp(x_1 - x_{-1}) + \exp(x_1 - x_0) = 2\exp(x_1)$. When using the change of coordinates we have the initial condition of:

$$\begin{aligned} x_2 &= x_2 \\ z_2 &= x_2 - x_0 = x_2 \\ y_2 &= x_2 - x_1 \\ u_2 &= x_3 + x_2 = 2\exp(x_1). \end{aligned} \tag{5.3}$$

So, search plans satisfying the recursion are parametrized by: $x_1 = s$ and $x_2 = t$. Hence, in the phase space

initial turning points are given by the surface:

$$\begin{aligned} x &= t \\ z &= t \\ y &= t - s \\ u &= 2\exp(s) = 2\exp(x - y) \end{aligned}$$

In the one ray case, candidates for optimal initial points, were the points that satisfied the initial condition and were contained on the separatrix which was an invariant curve. So, for the two ray case, we are interested in an invariant curve which contains points satisfying (5.3).

### 5.1.1 Numerical Evidence for Optimal Initial Tuning Points

The goal of this section is to find an invariant curve on which optimal turning points lay. It can be checked that iterating an arbitrary curve in the phase space with the inverse map (5.2) often yields iterations that are undefined after a few iterations. However, we can make some careful estimates which yields a curve that can be iterated backwards. Suppose we have a curve, $\phi(x)$, parametrized by $(x, u_\phi(x), y_\phi(x), z_\phi(x))$. In the change of variables, (5.1), we have $u_n = x_n + x_{n+1}$ and $z_{n+1} = y_{n+1} + y_n$. So, a reasonable estimate would be that, $u_\phi(x) \backsim 2x$ and $z_\phi(x) \backsim 2y_\phi(x)$.

Recall in the one ray case, the invariant curve of interest was a function of $x$ equal to $\ln(x) + O\left(\frac{\ln(x)}{x}\right)$. Using this fact, a reasonable guess would be that $y_\phi(x) = O(\ln(x))$. So, using the guesses of $y_\phi(x) = O(\ln(x))$, $u_\phi(x) \backsim 2x$ and $z_\phi(x) \backsim 2y_\phi(x)$ with some tuning of parameters, yielded a curve which showed promising results. That curve being:

$$\begin{aligned} x &= x \\ u &= 2x \\ y &= .5\ln(2x) \\ z &= \ln(2x) \end{aligned} \tag{5.4}$$

Starting with $x \geq 150$, the inverse map, (5.2), applied to (5.4) was able to be iterated backwards 51 times before becoming undefined at some $x$ values. However, the smallest resulting $x$ value from the backwards iterations was approximately 23. Also, when $y$ is plotted against $x$, we see the iterations bounce around (See Figure 5.2). If we hope to find optimal initial turning points, we need a curve whose iterates are more "stable". Also, from Proposition 2.3.3, we know $x_1$ cannot be too large. So, we want iterations that are

relatively close to the origin.

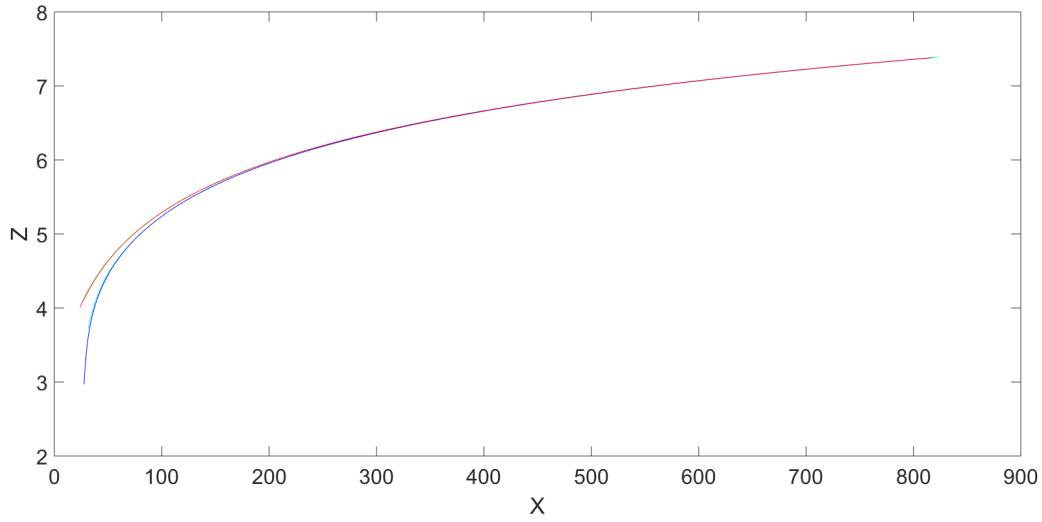


Figure 5.1: $z$ plotted against $x$ of (5.4) after 48-51 backwards iterations.

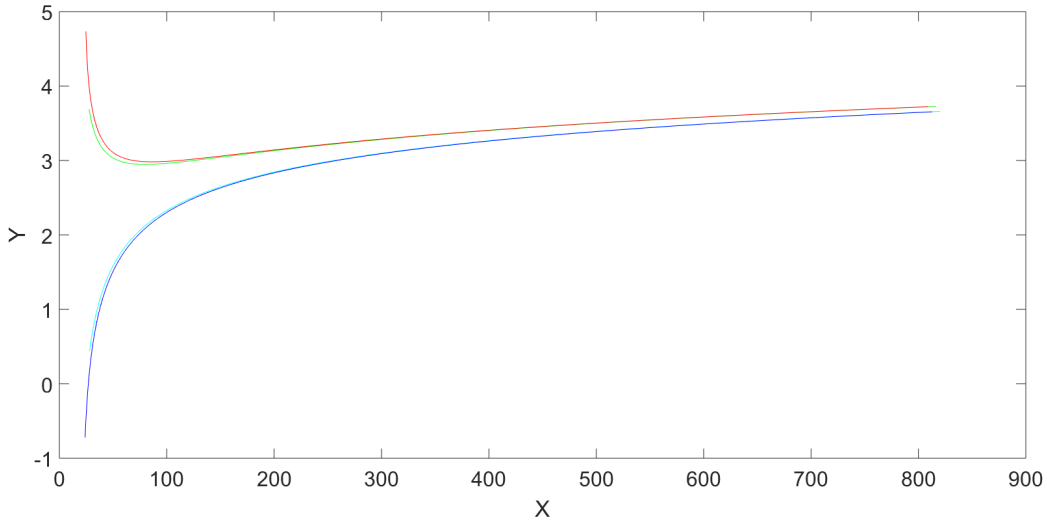


Figure 5.2: $y$ plotted against $x$ of (5.4) after 48-51 backwards iterations.

By adding additional parameters to (5.4), we can improve upon our initial guess. Adding $O\left(\frac{\ln(x)}{x}\right)$ terms to the parametrized curve and tuning parameters yielded iterations that converged very close the

origin.

$$\begin{aligned} x &= x \\ u &= 2x \\ y &= .5\ln(2x) + .25\ln(2x)/x \\ z &= \ln(2x) - .5\ln(2x)/x \end{aligned} \tag{5.5}$$

Starting with $x \geq 150$, the inverse map, (5.2), applied to (5.5) was able to be iterated backwards 67 times.

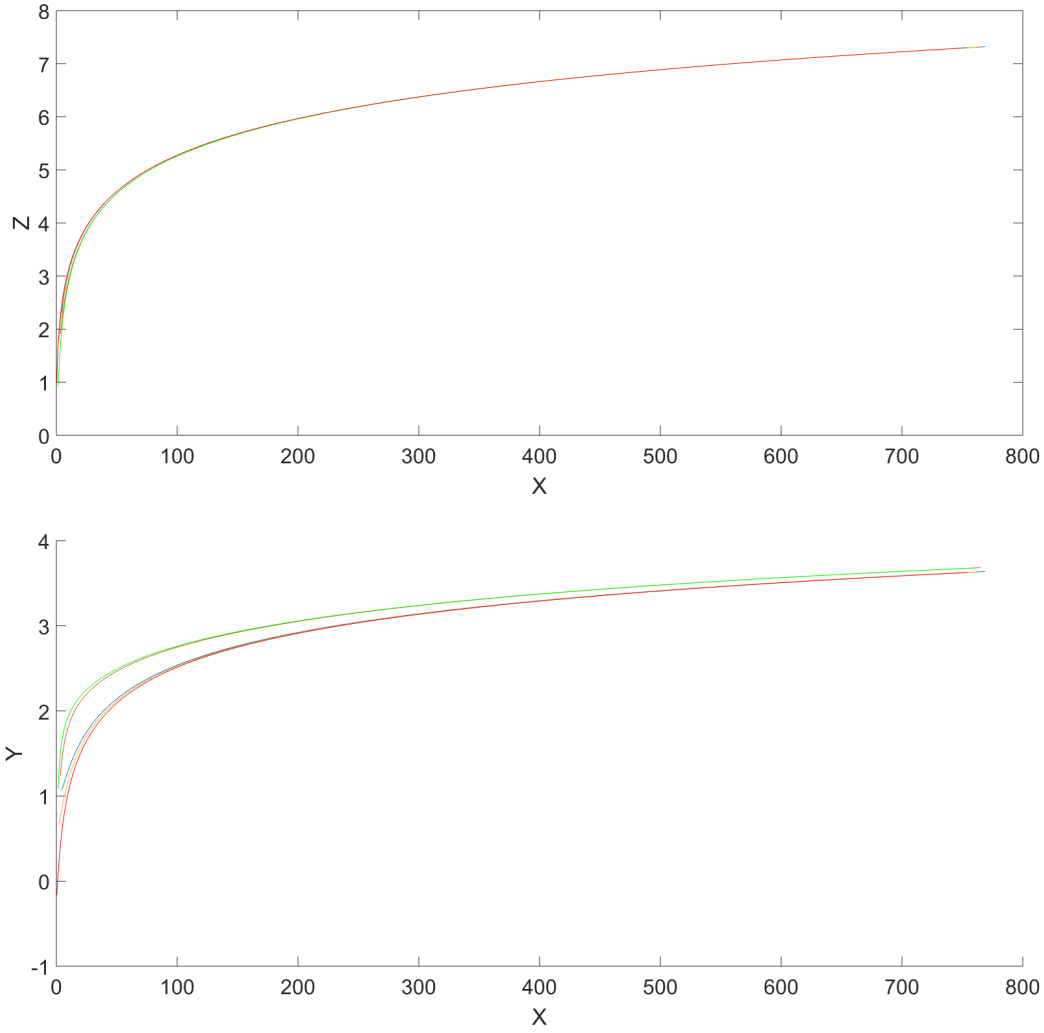


Figure 5.3: $z$ plotted against $x$ and $y$ plotted against $x$ of the line, (5.4), after 64-67 backwards iterations.

Now we have iterations that converge close to the origin but we also want the iterations to satisfy (5.3). However, backwards iterations of (5.5) did not satisfy (5.3). Through trial and error of adjusting parameters,

we are able to find a curve which has iterations that satisfy (5.3). That curve being:

$$\begin{aligned} x &= t \\ u &= 2t \\ y &= .5\ln(2t) + .25\ln(2t)/t \\ z &= \ln(2t) - .5\ln(2t)/t - .0003692 \end{aligned} \tag{5.6}$$

Backwards iterations of the curve, (5.6), yielded the figures below. Using these figures and the initial conditions (5.3), we can calculate the optimal initial turning points. When plotting $z$ against $x$, we see $z = x$ at $x \approx 1.1783$. (See Figure 5.4). So from the initial condition $z = x$, $x_2 \approx 1.1783$. Also, at $x \approx 1.1783$, $y \approx .6852$ and $u \approx 3.267$. Since $y = x_2 - x_1$, we have $x_1 \approx 0.4931$. Note that $2\exp(0.4931) = 3.27477... \approx 3.267$. Hence, $u_2 = 2\exp(x_1)$ from (5.3) is approximately satisfied.

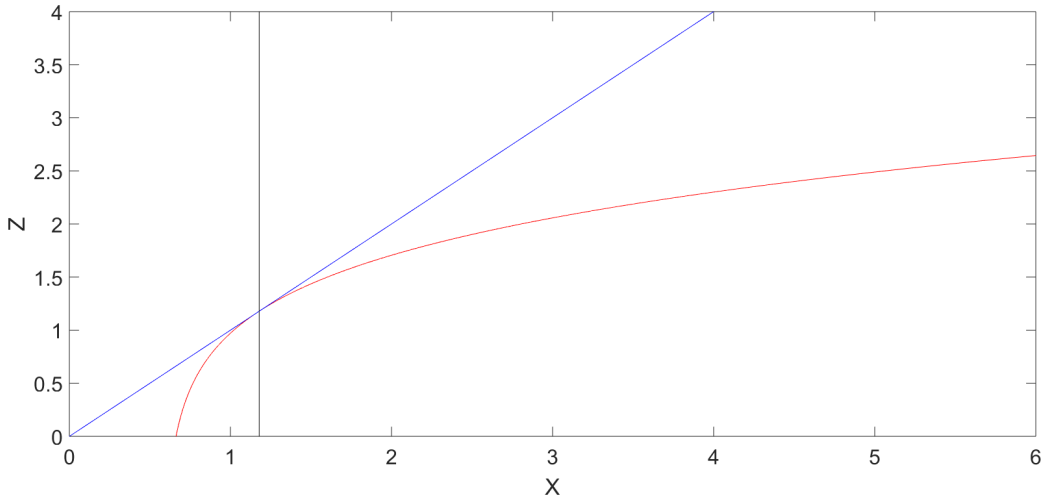


Figure 5.4: The curve represent the the iterated segment when $z$ is plotted against $x$. The curve intersects the line $z = x$ at $x = 1.1783...$

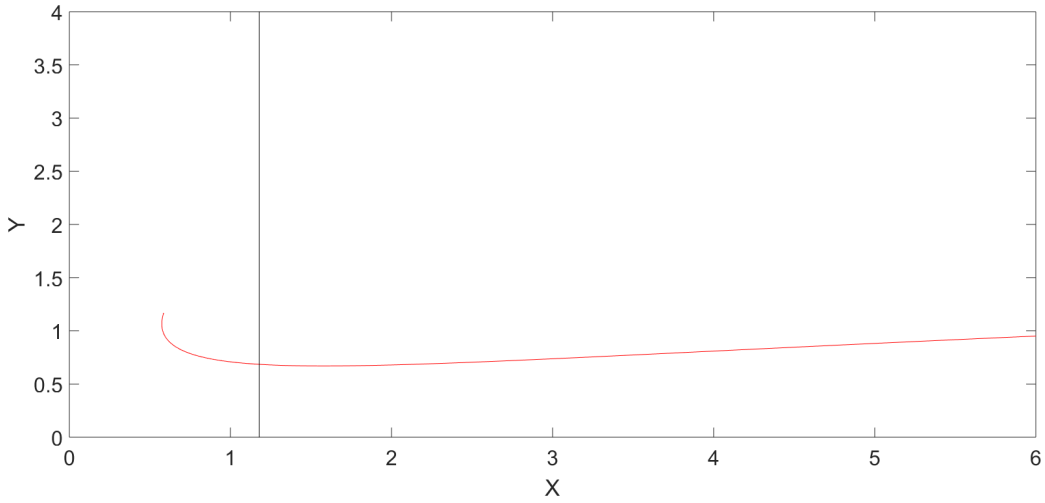


Figure 5.5: The curve represent the the iterated segment when $y$ is plotted against $x$. The curve intersects the line $x = 1.1783$ at $y = .6852$

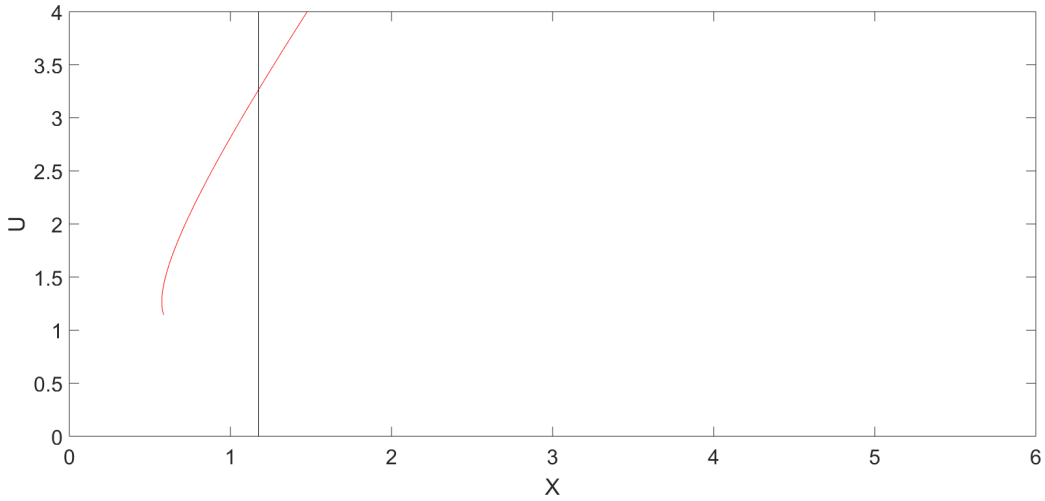


Figure 5.6: The curve represent the the iterated segment when $u$ is plotted against $x$. The curve intersects the line $x = 1.1783...$ at $u = 3.267...$

Forward iterations of $x_1$ and $x_2$ yield the search plan $\boldsymbol{x} \approx 0.4931, 1.1783, 2.0965, 3.1365, 4.3379, 5.5784, 7.1529, 7.7992, 13.7229,...$ with an expected cost of about 3.684.

It is not easy to find $x_1$ and $x_2$ such that forward iterations retain monotonicity. However, the results from this sections held monotonicity for eight iterations. Also, the minimum of 3.684 agrees with results from Chapter 4. The results and observations from this section lead to the following conjecture.

**Conjecture 5.1.1.** *For $f(x) = .5\exp(-x)$, there exists a one dimensional invariant curve, $\phi(x)$, where $\phi(x) = (x, u_\phi(x), y_\phi(x), z_\phi(x))$, in the phase space of (5.1) such that:*

- $u_\phi(x) = O(x)$, $y_\phi(x) = O(\ln(x))$ *and* $z_\phi(x) = O(\ln(x))$
- $u_\phi(x) \backsim 2x$ *and* $z_\phi(x) \backsim 2y_\phi(x)$
- *Optimal sequences are contained in* $\phi(x)$

## 5.2 Blindly Searching for Optimal Initial Tuning Points

The reason this section is called "Blindly Searching for Optimal Initial Tuning Points" is because we present a procedure that requires trial and error from the user but aids in finding optimal $x_1$ and $x_2$. Notice in Figure 5.2, the iterations "bounce" around. It seems like they are bouncing around an some invariant structure. This observation lead to the following idea. Iterate the initial guess backwards twice. Then average those two iterations and let that be the initial guess. Repeat this process. Then check if the curve generated from this process has backwards iterations which converge close to the origin. Below is an algorithm of the process.

### 5.2.1 Procedure

**Algorithm 5.2.1.**

Choose $(X, U, Y, Z)$, $N_1$ and $N_2$.

**while** $i \leq N_1$ **do**

Let $(x_3, u_3, y_3, z_3) = (X, U, Y, Z)$.

Using backwards iterations, calculate $(x_2, u_2, y_2, z_2)$ and $(x_1, u_1, y_1, z_1)$.

Let $(X, U, Y, Z) = .5((x_2, u_2, y_2, z_2) + (x_1, u_1, y_1, z_1))$.

$i = i + 1$

**end while**

$(x_{N_2}, u_{N_2}, y_{N_2}, z_{N_2}) = (X, U, Y, Z)$.

**for** $j = 1$ to $N_2$ **do**

Calculate $(x_{N_2-j}, u_{N_2-j}, y_{N_2-j}, z_{N_2-j})$.

**end for**

It takes some trial and error by the user to find the appropriate $(X, U, Y, Z)$, $N_1$ and $N_2$. Nevertheless, the method seems to work with multiple different $f(x)$. Now, we give an example of the procedure in action.

**Example 5.2.1.** Applying Algorithm 5.2.1 with $N_1 = 15$ and $N_2 = 46$ to (5.4) yielded the following graphs:

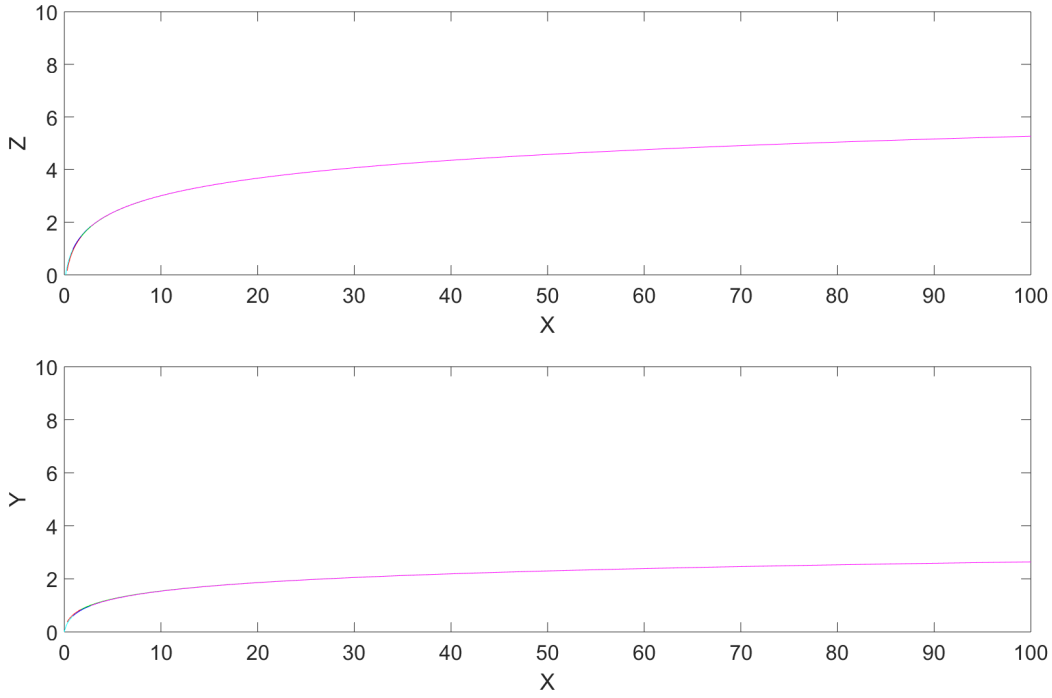


Figure 5.7: The last four iterations from Algorithm 5.2.1 with $N_1 = 15$, $N_2 = 46$ and (5.4)

The resulting iterations in Figure 5.7 are much closer than the iterations in Figures 5.2 and 5.3. Optimal sequences can be found using this output. To give an example other than $f(x) = .5\exp(-x)$ and to verify that the algorithm can be used to find optimal initial starting points, in the next subsection we use it to produce optimal starting points for $f(x) = .5\exp(-x^2)$.

### 5.2.2 Example

We use Algorithm 5.2.1 to numerically find the optimal starting points for $f(x) = .5\exp(-x^2)$. Let $f(x) = .5\exp(-x^2)$. Then the recurrence relation is given by:

$$x_{n+1} + x_{n+2} = \frac{\exp(x_n^2 - x_{n-1}^2) + \exp(x_n^2 - x_{n-2}^2)}{2x_n}$$

Using a change of variables yields:

$$\begin{aligned} x_{n+1} &= u_n - x_n \\ u_{n+1} &= \frac{\exp(y_n) + \exp(z_n)}{2x_n} \\ y_{n+1} &= x_{n+1}^2 - x_n^2 \\ z_{n+1} &= y_{n+1} + y_n \end{aligned} \tag{5.7}$$

The inverse map is given by:

$$\begin{aligned} x_n &= \sqrt{x_{n+1}^2 - y_{n+1}} \\ u_n &= x_n + x_{n+1} \\ y_n &= z_{n+1} - y_{n+1} \\ z_n &= \log(2x_n u_{n+1} - \exp(y_n)) \end{aligned} \tag{5.8}$$

Optimal sequences are defined by the initial choice of $x_1$ and $x_2$. Recall, $x_{-1}$ and $x_0$ are taken to be zero. So, we have $x_2 + x_3 = \dfrac{\exp(x_1^2 - x_{-1}^2) + \exp(x_1^2 - x_0^2)}{2x_n} = \dfrac{\exp(x_1^2)}{x_1}$. When using the change of coordinates we have:

$$\begin{aligned} x_2 &= x_2 \\ z_2 &= x_2^2 - x_0^2 = x_2^2 \\ y_2 &= x_2^2 - x_1^2 \\ u_2 &= x_3 + x_2 = \exp(x_1^2)/x_1 \end{aligned} \tag{5.9}$$

So, search plans satisfying the recursion are parametrized by: $x_1 = s$ and $x_2 = t$. Hence, in the phase space initial turning points are given by the surface:

$$\begin{aligned} x &= t \\ z &= t^2 \\ y &= t^2 - s^2 \\ u &= \exp(s^2)/s \end{aligned} \tag{5.10}$$

Using the algorithm with $N_1 = 752$, $N_2 = 990$, $u = 2x$, $y = 4\ln(\ln(x))$ and $z = 8\ln(\ln(x))$ generated the results below.

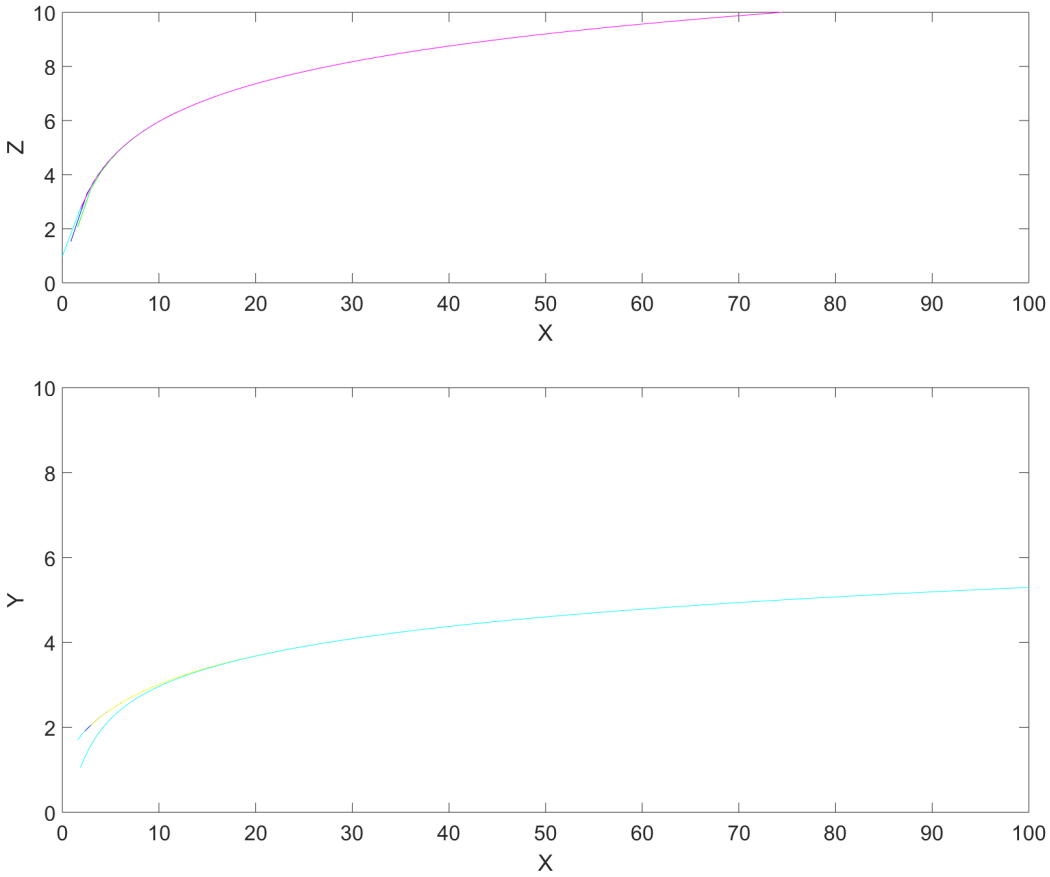


Figure 5.8: $z$ plotted against $x$ and $y$ plotted against $x$ for the last four iterations after using the algorithm

From the initial condition, (5.9), we need $z = x^2$, which happens at $x \approx 1.565$. (See Figure 5.9). So, $x_2 \approx 1.565$. Also, at $x \approx 1.565$, $y \approx .9540$ and $u \approx 3.639$. Since $y = x_2^2 - x_1^2$, we have $x_1 \approx 1.2214$. Note that $\exp(1.2214^2)/1.2214 = 3.63941... \approx 3.639$. Hence, $u_2 = \exp(x_1^2)/x_1$ from (5.9) is satisfied. There was no reason in particular the parameters in this example were chosen. Other parameters with tuning also generated similar results.

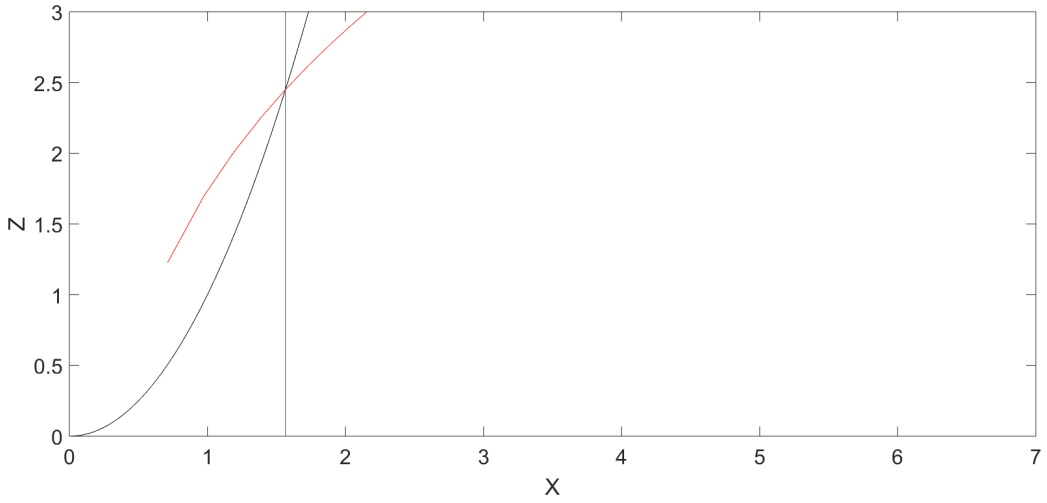


Figure 5.9: The curve represent the the iterated segment when $z$ is plotted against $x$. The curve intersects the line $z = x^2$ at $x = 1.5655...$

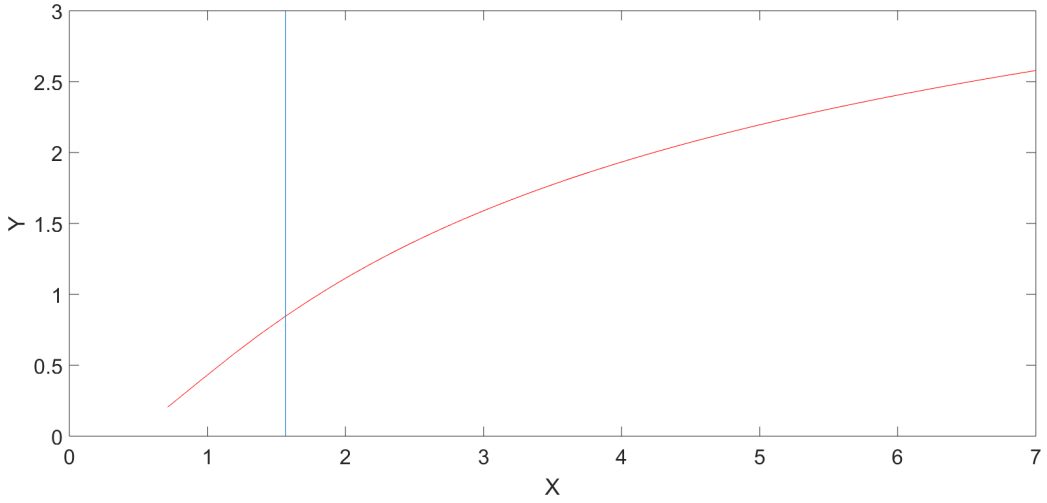


Figure 5.10: The curve represent the the iterated segment when $y$ is plotted against $x$. The curve intersects the line $x = 1.5655$ at $y = .9540...$

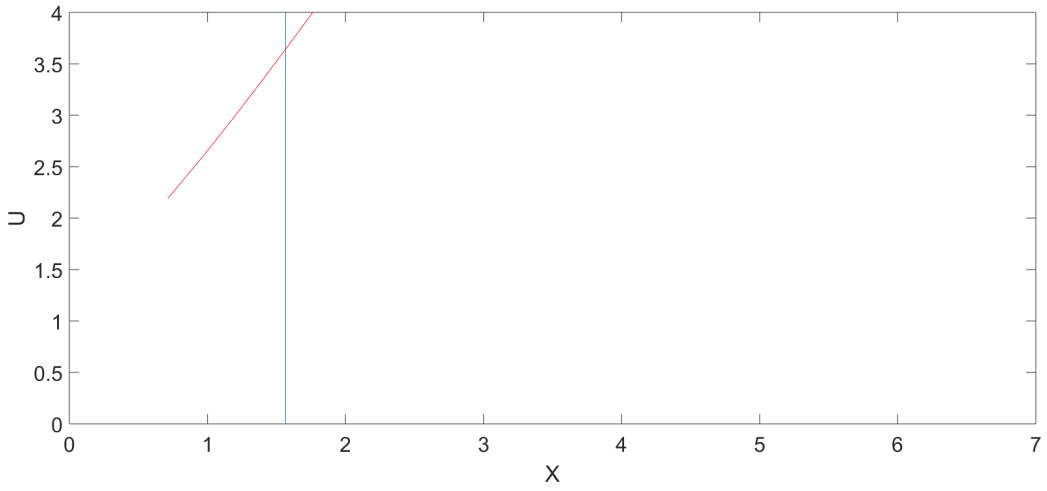


Figure 5.11: The curve represent the the iterated segment when $u$ is plotted against $x$. The curve intersects the line $x = 1.5655$ at $u = 3.639...$

Forward iterations of $x_1$ and $x_2$ yield the search plan $\boldsymbol{x} \approx 1.2214, 1.5655, 2.0739, 2.4634, 3.0718, 5.6792, 26.6922,...$ with an expected cost of approximately 2.6581. These results agree with results from Chapter 4. Although, the method in this subsection requires trial and error by the user, it aids in numerically finding optimal search plans.

# Chapter 6

# Closing Remarks and Future Directions

The linear search problem has had longevity and has persisted since 1963 because of its challenges and complexity. We investigated the linear search problem with low sensing on two rays in hopes of better understanding the linear search problem with low sensing on multiple rays. It was not necessarily clear that optimal sequences should be alternating or monotone between two rays. However, we were able to prove that optimal sequences must be monotone. We were also able to show that for a large class of distributions, optimal sequences must also be alternating between rays. These two properties may have seemed secondary at first but, as we saw, they are quite significant.

In Chapter 4, an algorithm was developed that estimated the minimum expected cost for any distribution to any desired accuracy. Although the algorithm did not take into account the underlying recurrence relation, it generated initial turning points close to the actual initial turning points of an optimal sequence. We also looked into the underlying dynamics for the blind linear search problem on two rays. In Chapter 5, we were able to show evidence of an invariant curve were optimal search plans lay. Using the underlying recurrence relation, we were able to numerically find $x_1$ and $x_2$ that generated optimal sequences for $f(x) = .5\exp(-x)$ and $f(x) = .5\exp(-x^2)$.

As some results of the one ray cases extend to the two ray case, it reasonable to assume that results from two ray would extend to multiple rays. Specifically, properties on the probability distribution that require optimal sequences to be monotone and alternating between rays. For multiple ray case, if it could be shown that optimal sequences must be alternating and monotone, then it should be rather straightforward to extend the dynamic program from the two ray case to multiple ray case. This would be a powerful tool because it would allow for verification of other numerical methods and vice-versa.

Another possibility for dynamic programming could be to develop an algorithm that considers both alternating and non-alternating search plans. With an algorithm that considers alternating and non-alternating search plans, one could numerically verify or disprove whether or not optimal search plans are always alternating for all multimodal distributions. However, optimal alternating search plans are monotone and optimal non-alternating search plans are not monotone (See Corollary 3.2.2). Monotonicity was highly significant

when developing the dynamic programming algorithm for alternating search plans in Chapter 4. So, no longer being able to assume monotonicity when developing an algorithm that considers both alternating and non-alternating search plans may be challenging.

Originally, I had set out prove analytically that there would exist a separatrix in the phase space that separated points that generated monotone sequences from those that did not. However, this turned out be rather difficult. In the one ray case, the phase space is two dimensional but in the two ray case the phase space is four dimensional. Since a curve can separate a two dimensional space, it could be the case that for the one ray case a separatrix existed by happenstance. However, a curve cannot be a candidate for a separatrix in a four dimensional space. If a separatrix did exist for the two ray case, it would be reasonable to assume optimal initial turning points would be contained in it. Hence, knowing whether or not a separatrix existed in the two ray case would be of high interest.

Another interesting question that I looked into but to no avail was: Do optimal turning points from the one ray case relate to optimal turning points from the two ray case? For example, would knowing the optimal $x_1$ for $f(x) = \exp(-x)$ on one ray shed any light on the optimal $x_1$ for $f(x) = .5\exp(-x)$ on two rays. If the optimal initial turning points did relate to each other, maybe it could be the case that the blind linear search problem on two rays could be mapped to two blind linear search problems on one ray. Since finding invariant curves and doing numerical calculations for the one ray case is much easier, being able to relate optimal turning points from the one ray case to optimal turning points from the two ray case would be a significant finding.

# References


[1] Alpern, Steve, and Shmuel Gal. The theory of search games and rendezvous. Vol. 55. Springer Science & Business Media, 2006.

[2] Beck, Anatole. "On the linear search problem." Israel Journal of Mathematics 2, no. 4 (1964): 221-228.

[3] Beck, Anatole, and D. J. Newman. "Yet more on the linear search problem." Israel journal of mathematics 8, no. 4 (1970): 419-429.

[4] Beck, Anatole, and Micah Beck. "Son of the linear search problem." Israel Journal of Mathematics 48, no. 2-3 (1984): 109-122.

[5] Beck, Anatole, and Micah Beck. "The linear search problem rides again." Israel Journal of Mathematics 53, no. 3 (1986): 365-372.

[6] Bellman, Richard. "An optimal search." SIAM review 5, no. 3 (1963): 274.

[7] Bruss, F. T., and J. B. Robertson. "A survey of the linear-search problem." Math. Scientist 13 (1988): 75-89.

[8] Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms-Secund Edition. McGraw-Hill, 2001.

[9] De Pablo, Irene Ruano, Aaron Becker, and Timothy Bretl. "An optimal solution to the linear search problem for a robot with dynamics." In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pp. 652-657. IEEE, 2010.

[10] Franck, Wallace. "An optimal search problem." SIAM review 7, no. 4 (1965): 503-512.

[11] Rousseeuw, Peter J. "Optimal search paths for random variables." Journal of Computational and Applied Mathematics 9, no. 3 (1983): 279-286.

[12] Washburn, Alan. "Dynamic programming and the backpacker's linear search problem." Journal of Computational and Applied Mathematics 60, no. 3 (1995): 357-365.

[13] Baryshnikov, Y., E. Coffman, P. Jelenkovi, P. MomIlovi, and Dan Rubenstein. "Flood search under the California split rule." Operations Research Letters 32, no. 3 (2004): 199-206.

[14] Baryshnikov, Yu, and Vadim Zharnitsky. "Search on the brink of chaos." Nonlinearity 25, no. 11 (2012): 3023.

# Appendix A

# Dynamic Program on One Ray

Since the main focus of this paper was the blind linear search problem on the real line, we give a dynamic program for the half-line case in the appendix. In this section we present a dynamic program that estimates the blind linear search problem on one ray. As with the real line case, the dynamic program only solves for the minimum expected cost with a discrete distribution on a finite set of points. However, it can be used to estimate the minimum expected cost for search plans with continuous distributions up to any desired accuracy. Proving that the algorithm can be used to estimate the blind linear search problem on one ray with any distribution is very similar to proving it for the real line case, which was presented in Chapter 4. So, we only the present the algorithm and give several examples to showcase its accuracy. We then compare the results from the algorithm to known results from [14].

For the one ray case, $E(\boldsymbol{x})$ also has the same scale invariance property that was stated in Remark 4.1.1. So, we can assume our probability mass distribution, $p(x)$, only take values on the points $0, 1, 2, ...n$. After using the the dynamic program, the minimum cost and turning points can be scaled appropriately.

Let $\boldsymbol{l}$ be the associated random variable for the probability mass function, $p(x)$. Let $f(x) = P(\boldsymbol{l} > |x|)$ and let $E^*$ denote the minimum expected cost. Define $H(i, j)$ to denote the minimum expected remaining cost given that the interval $[0, i]$ has been searched and the searcher is on an excursion at point $j$. When the searcher is at point $j$, it has two options to get to point $j + 1$. Either continue its current excursion by moving right to point $j+1$ or begin a new excursion by returning to the origin to determine if the object has been discovered then going to the point $j + 1$. If the searcher chooses the former, then to move right from $j$ to $j + 1$ costs an expected additional $1 * f(i)$. So, $H(i, j)$ would take on the value $1 * f(i) + H(i, j + 1)$. (See Figure A.1 and A.2). On the other hand if the searcher goes from point $j$ to the origin then to $j + 1$, the expected additional cost of the move would be $(j + 1) * f(j)$. So, $H(i, j)$ would take on the value $(j + 1)f(j) + H(j, j + 1)$. (See Figure A.1 and A.3). For boundary conditions, if the searcher is at point $n$ then there is no where else for it to go once it has verified the item has been collected. So, $H(i, n) = 0$. Lastly, if the searcher is at the origin and has not searched any interval, its only choice would be to move right one to point 1. Thus $H(0, 0)$ takes on the value $1 + H(0, 1)$. Putting this all together yields:

- $H(i, n) = 0$
- $H(i, j) = \min\{f(i) + H(i, j + 1), (j + 1)f(j) + H(j, j + 1)\}$
- $E^* = H(0, 0) = 1 + H(0, 1)$

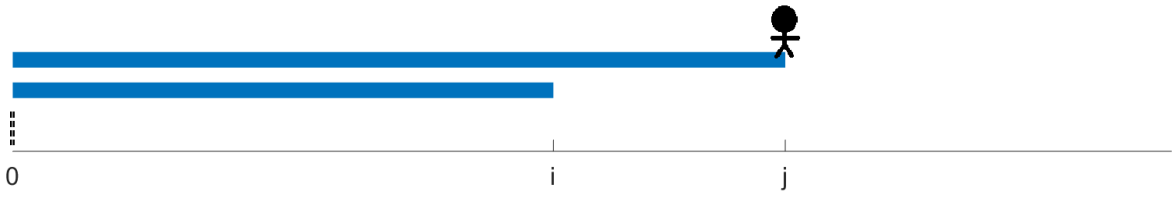


Figure A.1: $H(i, j)$

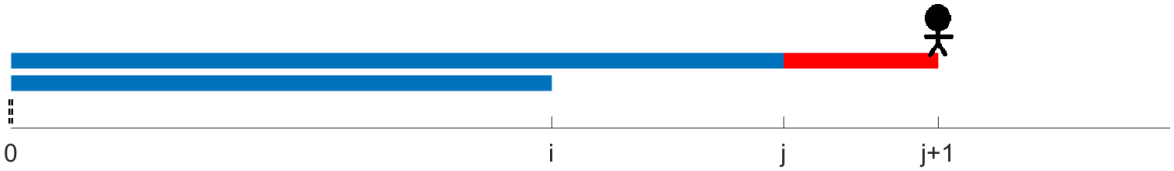


Figure A.2: The searcher continues its current excursion by going to $j + 1$ from $j$. It takes the searcher an expected additional $1 * f(i)$ to get to $H(i, j + 1)$ from $H(i, j)$.

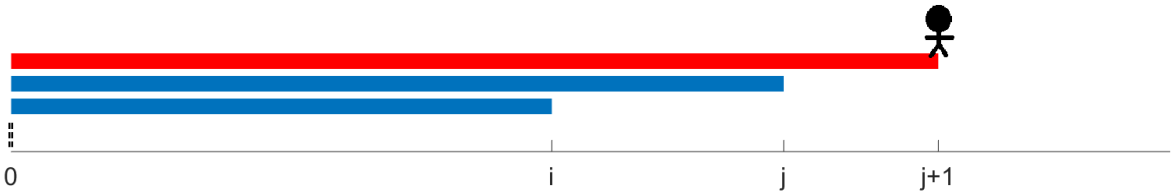


Figure A.3: The searcher begins a new excursion by going to 0 from $j$, then $j+1$ from 0. It takes the searcher an expected additional $(j+1)*f(j)$ to get to $H(j,j+1)$ from $H(i,j)$.

Having to check all possible search plans on the set $\{0,1,2,...,n\}$ to find the minimum expected cost would take on the order of $O(2^n)$ computations. However, to calculate all the $H(i,j)$'s for the dynamic programming would only take on the order of $O(n^2)$ computations. This is because all the $H(i,j)$'s can be calculated and stored in an $n \times n$ array. Not only are the computations relatively fast, the dynamic programming is also powerful to verifying results when developing other numerical methods for the linear search problem.

## Examples

In this section we give examples of the dynamic programming used to estimate the minimum expected value for several different $f(x)$. The first two functions, $\exp(-x)$ and $\exp(-x^2)$, were chosen because numerical solutions were shown in the paper [14]. We will give a brief overview of how these initial turning points were numerically calculated. Then we will use the results to verify that the dynamic program generates accurate estimates of the minimum expected cost. The third function, $f(x) = 1 - \log(x+1)/\log(2)$, was chosen because the minimum expected cost can be easily calculated by hand. In all three scenarios, not only were the estimated expected minimum costs close in numerical values but also the estimated initial turning points.

**Example A.0.1.** For $f(x) = \exp(-x)$, details of how the separatrix was calculated were given Section 1.2. In the phase space, the straight line of initial turning points intersects the separatrix at two points

$x \approx .7465...$ and $x \approx .1954...$ (Figure 1.2). The initial turning point $x_1 = .7465...$ generated the lower expected cost of $E^* \approx 2.3645$. The dynamic program with $f(x) = \exp(-x)$ on the interval $[0, 12]$ and an .012-grid estimated the expected minimum cost to be 2.3645 and the first turning point to be .744.

**Example A.0.2.** When $f(x) = \exp(-x^2)$ the recurrence relation is given by:

$$x_{n+1} = \frac{1}{2x_n} \exp(x_n^2 - x_{n-1}^2).$$

Using a change of variable, the recursion can be rewritten as:

$$x_{n+1} = \frac{1}{2x_n} \exp(y_n)$$
$$y_{n+1} = x_{n+1}^2 - x_n^2$$

The inverse map of the above equations is given by:

$$x_n = \sqrt{x_{n+1}^2 - y_{n+1}}$$
$$y_n = \ln(2x_{n+1}x_n)$$

Applying the inverse map to boundary of $y > 0$ yields iterations that converge to the separatrix. Since $x_0 = 0$, initial turning points are generated by the curve $y_1 = x_1^2 - x_0^2 = x_1^2 - 0^2 = t^2$.

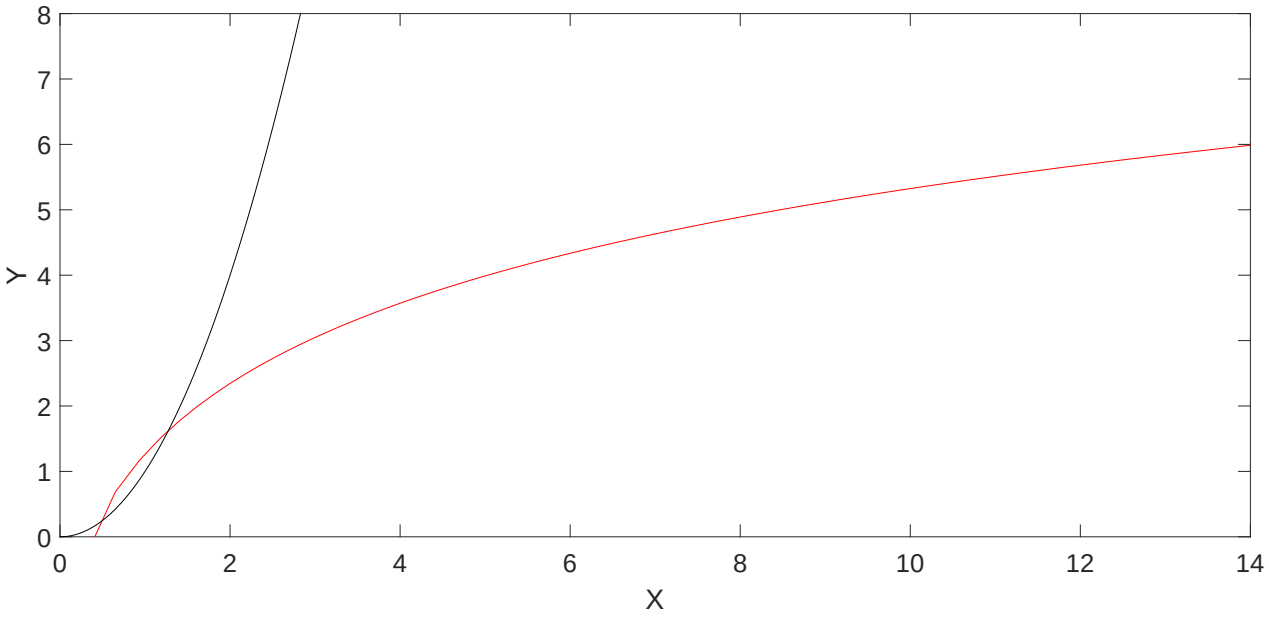


Figure A.4: The parabola passing through the origin represents the curve $y = x^2$ and points on at that parabola are candidates for initial turning points. The other curve represents the separatrix.

The optimal initial turning point from Figure A.4 is $x_1 \approx 1.2738$. Which yielded a minimum expected cost of about 1.7198. The dynamic program with $f(x) = \exp(-x^2)$ on the domain $[0, 9]$ with a .009-grid, generated an initial turning point of 1.278 and a minimum expected cost of approximately 1.72002.

**Example A.0.3.** Although, $f(x) = 1 - \log(x+1)/\log(2)$ is an odd function to consider. It is nice in the fact that the optimal solution can be checked by just paper and pencil. Since $f'(1) \neq 0$, an optimal search plan for $f(x)$ will have only finitely many turning points (Refer to Proposition 2.1.3). The turning points $x_1 = \dfrac{1}{\log(2)} - 1 \approx .44270...$ and $x_2 = 1$ satisfy the recursion $f(x_0) + x_2 f'(x_1) = f(0) + x_2 f'(x_1) = 0$. The minimum expected cost yielded by these points is approximately .913928. The dynamic program with a .001-grid on $[0, 1]$ gives an expected minimum cost of approximately .913928 and turning points of $x_1 = .443$ and $x_2 = 1$.