International Series in Operations Research & Management Science

Lawrence D. Stone
Johannes O. Royset
Alan R. Washburn

Optimal Search for Moving Targets

EXTRAS ONLINE

Springer

International Series in Operations Research & Management Science

Volume 237

More information about this series at http://www.springer.com/series/6161

Lawrence D. Stone • Johannes O. Royset
Alan R. Washburn

Optimal Search for Moving Targets

Springer

Lawrence D. Stone
Metron, Inc.
Reston, VA, USA

Johannes O. Royset
Naval Postgraduate School
Monterey, CA, USA

Alan R. Washburn
Naval Postgraduate School
Monterey, CA, USA

ISSN 0884-8289 ISSN 2214-7934 (electronic)
International Series in Operations Research & Management Science
ISBN 978-3-319-26897-2 ISBN 978-3-319-26899-6 (eBook)
DOI 10.1007/978-3-319-26899-6

Library of Congress Control Number: 2016931331

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Foreword

It has been more than 40 years since the publication in 1975 of the definitive book, *Theory of Optimal Search*, which dealt almost exclusively with the stationary target search problem. Since then the theory has advanced to encompass search for targets that move even as the search proceeds, and computers have developed sufficient capability to employ the improved theory. The problem of how to search for moving targets arises every day in military, rescue, law enforcement, and border patrol operations. In this book the authors document and explain this expanded theory of search and show how it can be applied.

Shortly after 1975, Scott Brown, Alan Washburn, Lawrence Stone, and others began the development of a theory for solving optimal search for moving target problems along with algorithms for computing optimal search plans. This theory is applicable to problems where the target does not react to or anticipate the efforts of the searcher. It assumes that search effort can be spread over the search area without constraint except on the total amount of effort. When the searcher's path is constrained so that where search is applied at one time restricts where it can applied at the next, these results do not apply. For years progress on this NP-complete class of constrained searcher path problems was slow and difficult; the ability to solve these problems was limited by the daunting computation complexity of finding solutions. Recently, Johannes Royset and others have taken advantage of advances in solving mixed-integer programs and optimal control problems to formulate and solve constrained searcher path problems that involve multiple searchers and targets as well as realistic operational constraints. Chapters 1, 2, 3, 4, 5, and 6 collect and present the results of the work since 1975 in an accessible fashion with examples that demonstrate how to apply the results to realistic problems.

This book begins with a review of basic results in optimal search for a stationary target. It then develops the theory of optimal search for a moving target, providing algorithms for computing optimal plans and examples of their use. Next, it develops methods for computing optimal search plans involving multiple targets and multiple searchers with realistic operational constraints on searcher movement. These results all assume that the target does not react to the search. In the final chapter, there is a brief overview of mostly military problems where the target tries to avoid being found as well as rescue or rendezvous problems where target and searcher attempt to cooperate.

Acknowledgments

The authors would like to thank Stephen L. Anderson for producing Fig. 2.5 and the examples in Chaps. 3 and 5. He also wrote the MATLAB code included in this book. This code can be used to reproduce the examples in Chaps. 3 and 5 and as a basis for finding optimal search plans for problems based on those examples.

We further express our appreciation to the people who read and commented on drafts of our manuscript, in particular, David Hall, The Pennsylvania State University, and Barry Belkin, Daniel H. Wagner Associates.

For the second author, financial support over many years from the Office of Naval Research, Air Force Office of Scientific Research, and the Naval Postgraduate School – Special Operations Command Field Experimentation Program played a critical role in the development of the results behind parts of this book. He would like to thank David Netzer for allowing us to bring search theory to flight operations with unmanned aircrafts at Camp Roberts, CA; Michael Atkinson, Hoam Chung, Qi Gong, Isaac Kaminer, Moshe Kress and, Elijah Polak for invigorating collaboration; and the numerous students at the Naval Postgraduate School and University of California Santa Cruz for their contributions to search theory. In particular, the work with Joseph Foraker, Chris Phelps, and Hiroyuki Sato was essential for this book.

Contents

About the Authors

Lawrence D. Stone is chief scientist at Metron Inc. He is a member of the National Academy of Engineering and a fellow of the Institute for Operations Research and Management Science.

In 1975, the Operations Research Society of America awarded the Lanchester Prize to Dr. Stone's text, *Theory of Optimal Search*. In 1986, he produced the probability maps used to locate the *S.S. Central America* which sank in 1857, taking millions of dollars of gold coins and bars to the ocean bottom one and one-half miles below. In 2010 he led the team that produced the probability distribution that guided the French to the location of the underwater wreckage of Air France Flight AF447.

He is a coauthor of the 2014 book, *Bayesian Multiple Target Tracking*. He continues to work on a number of detection and tracking systems for the US Navy and Coast Guard including the Search and Rescue Optimal Planning System used by the Coast Guard since 2007 to plan searches for people missing at sea.

Education	Bachelor of Science, Mathematics, Antioch College, 1964
Education	Master of Science, Mathematics, Purdue University, 1966
Education	Doctor of Philosophy, Mathematics, Purdue University, 1967

Johannes O. Royset is associate chair of Research and associate professor of Operations Research at the Naval Postgraduate School. Dr. Royset's research focuses on formulating and solving stochastic and deterministic optimization problems arising in data analysis, sensor management, and reliability engineering. Dr. Royset has a doctor of philosophy degree from the University of California at Berkeley (2002). He was awarded a National Research Council postdoctoral fellowship in 2003, a Young Investigator Award from the Air Force Office of Scientific Research in 2007, and the Barchi Prize as well as the MOR Journal Award from the Military Operations Research Society in 2009. He received the Carl E. and Jessie W. Menneken Faculty Award for Excellence in Scientific Research in 2010. Dr. Royset is an associate editor of *Operations Research*, *Naval Research Logistics*, *Journal of Optimization Theory and Applications*, and *Computational Optimization and Applications*.

Alan R. Washburn is Professor emeritus at the Naval Postgraduate School. He received a Ph.D. in electrical engineering from Carnegie Institute of Technology in 1965 and has been with the Operations Research Department at the Naval Postgraduate School since 1970. He is a member of the National Academy of Engineering and is the recipient of several awards and prizes, including the Distinguished Civilian Service Award for his research and tutorial notes on several topics of importance to the Department of Defense. He is the author of over 50 scientific publications, including several books. His research is almost entirely military, emphasizing problems that employ search theory, game theory, or both. He is a member of INFORMS and also of the Military Operations Research Society (MORS).

Chapter 1
Introduction

The problem of how to search for a moving target[1] arises every day. Search problems arise in military, rescue, law enforcement, and border patrol operations. In military operations, the searchers may be aircraft looking for suspected individuals or downed pilots in an area of interest. The U. S. Navy has a long history of planning searches for adversarial submarines. Park rangers may search for lost hikers. Almost every day someone is lost in a wilderness or rural area, and volunteer search and rescue groups plan and execute searches to find them – Koester (2008). In a damaged or burning building, fire fighters and ground robots may search for trapped individuals. Law enforcement officers may act as searchers when looking for criminals. Near national borders, the searchers may be border patrols seeking illegal immigrants. The searchers may also be Coast Guard cutters and helicopters scanning the ocean for smugglers.

The development of a formal theory of search was begun by Bernard O. Koopman and his colleagues during World War II in the U.S. Navy's Operations Evaluation Group. These results were collected and summarized in Koopman (1946) which was initially a classified document. Koopman (1956a, b, 1957) published a series of papers based on Koopman (1946) that presented the mathematical foundations of search theory. In subsequent years, the theory was expanded and extended to a wide range of problems involving both stationary and moving targets. See Benkoski et al. (1991) for a survey of this work. By now this theory has been embodied in various

[1]We refer to the search object as the target even though it may be benign, for example a person lost in the woods.

L.D. Stone et al., *Optimal Search for Moving Targets*, International Series in Operations Research & Management Science 237,
DOI 10.1007/978-3-319-26899-6_1

software suites for search plans and operations. For example the U. S. Coast Guard regularly uses the Search And Rescue Optimal Planning System (SAROPS) to plan searches for people and boats missing at sea – Kratzke et al. (2010).

Examples There have been a number of high-profile searches for stationary and moving targets. The following are examples in which formal search planning played an important role:

- The nuclear submarine, USS *Scorpion*, was lost in May 1968 on its way from the Azores to its home port of Norfolk, Virginia. The search for *Scorpion* lasted from May until October 1968 when the submarine was found on the ocean bottom. The methods used to plan this search are discussed in Richardson and Stone (1971).
- The *SS Central America*, a ship carrying gold and passengers from San Francisco to New York, sank in 1857. In 1988 after 2 years of searching the ocean bottom, the ship was found at a depth of 8000 ft. In 1989 1 t of gold was recovered from the wreck. See Stone (1992).
- Air France flight, AF 447, disappeared in the early morning hours of June 1, 2009 in a remote part of the Atlantic Ocean near the equator. The underwater wreckage was found on April 3, 2001 almost 2 years after the plane disappeared. See Stone et al. (2014).
- SAROPS was used to plan the search for a fisherman who fell overboard 40 miles off the tip of Long Island in the early morning hours of July 24, 2013. He drifted for almost 12 h before he was found alive and rescued by a Coast Guard helicopter. See Tough (2014).
- Using search planning software based on the methods described in Chap. 3, the U. S. Navy doubled its success rate when searching for Soviet submarines during the cold war. Unfortunately, documentation of these searches is classified.

The following are two unsuccessful searches that might have benefitted from better planning:

- Steve Fossett disappeared on September 3, 2007 in a remote area of Nevada during a solo flight in a small aircraft. An intense search conducted by both private parties and multiple government agencies was unsuccessful. Fossett's wreck was discovered in the eastern Sierra Nevada mountains by a hiker almost 1 year later. Keller (2009) describes an independent effort to direct search efforts using search theory and optimization techniques and discusses why it was not successful.
- On July 1, 1986, 9-year old Andrew Warburton left his relatives' house in Nova Scotia, where his parents were staying, to join friends at a swimming hole in the woods a short distance from the house. After an extensive search he was found dead from exposure on July 8 about 4 km from the house. This search is a tragic example of poor and unsystematic search planning. See the Montreal Gazette (1986).

Expectations In this book the reader will find derivations of the main results in optimal search for moving targets and algorithms for computing optimal plans. Examples of using these algorithms to find optimal plans are presented along with discussions of numerical complexity, references to tools and software, and links to MATLAB code that implements some of the algorithms.

1.1 Search Problem

Finding the optimal search plan for a moving target often relies on being able to find optimal search plans for stationary targets, so in Chap. 2 we begin with the problem of search for a stationary target. We formulate the problem in Bayesian terms. For a stationary target there is a prior distribution on the target's location. We usually think of the target's location as being somewhere in a region of two or three dimensional space or in a set of cells. For the moment let's consider the situation where the target is located in one of a set of cells. We have a sensor (e.g., our eye) that can detected the target and a certain amount of search effort (e.g., time) available to detect the target. There is a detection function that relates the amount of time that we search (look) in a cell to the probability of detecting the target given it is in that cell. The basic stationary search problem is to allocate the available search effort among the cells to maximize the probability of detecting the target.

For a moving target, we specify a probability distribution on the target's initial location and provide a probabilistic description of how the target moves through space. This determines the target's motion model, i.e., the probability distribution on the set of possible target paths over the time period of interest. Let T be the time available for search. At each time $t = 0, \ldots, T$, the target is located in a one of the cells, and just before time $t + 1$, it can move to another cell as determined by the target motion model. We have a limited amount of search effort available at each time. The basic moving target search problem is to find an allocation of search effort in space and time that maximizes the probability of detecting the target by time T while satisfying the search effort constraints.

Except for Chap. 7, we assume that the target's motion is independent of the search effort. The target can move, but it does not react to the search. This is called a one-sided search problem. In addition, Chaps. 2, 3 and 5 assume that the placement of search effort is unconstrained. For example, if search is placed in a cell at time t, the search at time $t + 1$ can be in any cell in the search space. If search effort is continuous, it is infinitely divisible over the search space at no cost to the searcher. The case where the placement of search effort at time t constrains where search effort can be placed at time $t + 1$ presents an interesting and difficult search problem called the constrained searcher path problem. This problem is addressed in Chaps. 4 and 6.

The following sections describe the contents of the chapters.

1.2 Chapter 2: Search for a Stationary Target

Optimal search for a stationary target is covered extensively in Stone (2007). Chapter 2 provides an overview of the basic results in this area. These results are sufficient to allow the reader to use and understand this book without having to refer to Stone (2007). This chapter defines the search space and prior distribution on target location in discrete and continuous space. It discusses search sensors and some basic notions of detection modeling, including lateral range curves, sweep widths, and detection functions. It develops methods for finding search plans that maximize the probability of detecting the target by a fixed time and explores related optimal search problems such as minimizing the mean time to find the target. It presents algorithms that may be used to compute optimal plans for stationary targets in most cases.

Stone (2007) extends the results presented in this chapter primarily in the generality of its results and by covering topics such as optimal search and stop, search in the presence of false targets, and approximation of optimal plans.

1.3 Chapter 3: Search for a Moving Target in Discrete Space and Time

The simplest moving target problems are those that take place in discrete space and time. The target state space is a set of J cells, for example a two-dimensional grid of cells. There is a probability distribution on the target's location at time $t = 0$. The target motion model specifies, in a probabilistic fashion, how the target moves from a cell at time t to a cell at time $t + 1$.

At each time $t = 0, \ldots, T$ we have a fixed amount $m(t)$ of search effort available. Effort may be available in discrete glimpses or in continuous amounts. A search allocation specifies the amount of search effort $f(j, t)$ to be spent in cell j at time t for $t = 0, \ldots, T$ and $j = 1, \ldots, J$.

This chapter presents necessary and sufficient conditions for a discrete-time-and-space search plan to maximize probability of detection by time T. Such plans are called T-optimal. When the detection function is exponential, the necessary and sufficient conditions can be stated in terms of optimal search for a stationary target as follows. If f^* is a T-optimal plan, then for each $t = 0, \ldots, T$, the allocation of effort at time t, $f^*(\cdot, t)$, is an optimal allocation of $m(t)$ effort for the stationary target problem whose distribution is obtained by conditioning on the failure of the search at all times other than t.

In this case one can find a T-optimal search plan by finding a sequence of optimal stationary target plans, and this chapter presents an efficient algorithm for doing this. In theory, the algorithm requires one to solve an infinite number of stationary target problems to construct moving target plans that converge to the T-optimal search plan. To alleviate this problem, this chapter shows how to compute an upper bound on how far from optimal the present plan in the sequence is. Usually, moving target plans obtained through solving this sequence of optimal stationary target plans converge quickly to a plan that is close to optimal.

The algorithm for finding T-optimal plans proceeds by going forward from time $t = 0$ to time $t = T$ finding optimal allocations for a stationary target at those times and constructing a moving target plan from them. It then goes back to $t = 0$ and repeats the process. We call this a Forward And Backward (FAB) algorithm. Chapter 3 finishes with a more general version of this FAB algorithm that applies to a range of search-related problems.

1.4 Chapter 4: Path Constrained Search in Discrete Time and Space

This chapter considers search problems where the targets move according to Markov or conditionally deterministic models and there are one or more searchers whose movements are constrained. In particular if a searcher is in cell j at time t, there is a restricted set $\mathcal{F}(j)$ to which it can move at time $t + 1$, i.e., the searcher's path is constrained. In the case of a single searcher and single target, the chapter presents a branch and bound algorithm for finding T-optimal plans. The chapter proceeds to more complex and operationally realistic problems where there are multiple searchers and multiple targets, for example, multiple drones looking for multiple targets. The searchers as well as the targets can have different characteristics. For this class of problems, a more general and flexible mathematical programming approach is developed. Using this approach, a number of examples are computed to illustrate the types of problems that can be solved by this method.

1.5 Chapter 5: Search for Moving Targets in Continuous Space

In Chap. 5, the search space is continuous, e.g., the plane. In continuous space, search plans are specified in terms of search density functions and probability distributions are specified by probability density functions. For discrete time, we find necessary and sufficient conditions for a T-optimal plan. When the detection function is exponential, we show that a necessary and sufficient condition for a search plan f^* to be T-optimal is that $f^*(\cdot, t)$ is an optimal allocation of $m(t)$ effort for the stationary target whose distribution is obtained by conditioning on the failure of the search at all times other than t. As in Chap. 3, this leads to an algorithm for finding T-optimal plans that proceeds by solving a sequence of stationary target problems which can be solved by the methods in Chap. 2.

The remainder of the chapter considers continuous-space search problems in continuous time and more general payoff functions than probability of detection.

It finds necessary and sufficient conditions for this more general class of payoff functions. It identifies, as special cases, all the payoff functions mentioned in Chap. 3 including minimizing mean time to detection and the class of FAB payoff functions.

1.6 Chapter 6: Constrained Search in Continuous Time and Space

In many search applications such as those involving the use of drones to search for moving targets such as smugglers trying to cross a border or the use of unmanned underwater vehicles to detect transiting submarines trying to penetrate a barrier, it is natural to consider search paths in continuous time and space. In most instances, the search platforms will have constraints on their speed, turn radius, and other performance parameters. In these cases we are faced with the problem of finding optimal search paths in continuous time and space subject to constraints. This chapter addresses this problem as an uncertain optimal control problem and develops approximate solutions that may be found using standard optimization solvers. This approach is illustrated with a number of examples involving one or more searchers and targets.

1.7 Chapter 7: Search Games

There are three possibilities for the attitude of the target of search. The first possibility is that the target does not care whether it is found or not, or does not know that a search is being conducted, or is not sentient. This is the most important of the three possibilities, and is covered in the previous chapters. A second possibility is that the target would prefer not to be found. This is often the case in military applications, and is the subject of Sect. 7.2. The applicable theory for this case is that of two-person zero-sum games. The third possibility is that the target would like to be found, and is aware that a search is being conducted for it. The target and the searcher now have the same goal, but are unable to communicate. This is the theory of rendezvous search that is summarized in Sect. 7.3.

References

Benkoski SJ, Monticino MG, Weisinger JR (1991) A survey of the search theory literature. Nav Res 38:469–494

Keller CM (2009) Applying optimal search theory to inland SAR: Steve Fossett case study. Paper presented at the 77th military operations research society symposium, Fort Leavenworth, Kansas

Koester RJ (2008) Lost person behavior. dBS Productions, Charlottesville

Koopman BO (1946) Search and screening, vol 56, Operations evaluation group report. Center for Naval Analyses, Alexandria

Koopman BO (1956a) The theory of search. Part I. Kinematic bases. Oper Res 4:324–346

Koopman BO (1956b) The theory of search. Part II. Target detection. Oper Res 4:503–531

Koopman BO (1957) The theory of search. Part III. The optimum distribution of searching effort. Oper Res 5:613–626

Kratzke TM, Stone LD, Frost JR (2010) Search and rescue optimal planning system. In: Proceedings of 13th international conference on information fusion (FUSION 2010), Edinburgh, Scotland, 26–29 July 2010

Montreal Gazette (1986) Boy lost a week in N. S. woods is found dead. The Montreal Gazette, 9 July, p. B1, Available via Google News at https://news.google.com

Richardson HR, Stone LD (1971) Operations analysis during the underwater search for *Scorpion*. Nav Res Logist 18:141–157

Stone LD (1992) Search for the SS *Central America*: mathematical treasure hunting. Interfaces 22:32–54

Stone LD (2007) Theory of optimal search, 2nd edn. INFORMS, Catonsville

Stone LD, Keller CM, Kratzke TM, Strumpfer JP (2014) Search for the wreckage of Air France AF 447. Stat Sci 29:69–80

Tough P (2014) Man overboard. New York Times Magazine, 5 Jan 2014

Chapter 2
Search for a Stationary Target

Many algorithms for computing optimal search plans for a moving target rely on being able to compute optimal plans for a stationary target. This chapter provides an overview of the standard models and results for optimal search for a stationary target. It discusses search sensors and some basic notions of detection modeling, including lateral range curves, sweep widths, and detection functions. It defines the search space and the prior distribution on target location in discrete and continuous space. It develops methods for finding search plans that maximize the probability of detecting the target by a fixed time and explores related optimal search problems such as minimizing the mean time to find the target. It presents algorithms that may be used to compute optimal plans in most cases. Optimal search for a stationary target is covered more extensively in Stone (2007).

2.1 Prior Distributions

We assume there is a prior distribution on the target's location. This distribution captures our knowledge and uncertainty about the target's location. Typically this distribution includes subjective as well as objective information whose uncertainties have been expressed in terms of probability distributions.

We usually think of the target's location as being somewhere in a region of two or three dimensional space or in a set of cells. First let's consider the situation where the target is located in one of a set of cells. The cells most often refer to physical locations, but that is not necessary.

L.D. Stone et al., *Optimal Search for Moving Targets*, International Series
in Operations Research & Management Science 237,
DOI 10.1007/978-3-319-26899-6_2

2.1.1 Discrete Prior Distributions

A discrete prior is a probability distribution on a set of J cells. For simplicity, we assume that the number of cells is finite and that the cells are numbered $j = 1, \ldots, J$. The cells can represent discrete locations or cells in a grid; they can represent any situation where the target can be in one of a finite set of locations. We use the term cell generically to mean any of these possibilities. A discrete prior specifies the probability $p(j)$ that the target is in cell j for $j = 1, \ldots, J$, and we assume

$$\sum\nolimits_{j=1}^{J} p(j) = 1.$$

Even when the target search space is continuous, it is often convenient to impose a grid of cells on the space and approximate the underlying target location distribution with a cellular one as was done in the search for the *SS Central America,* which sank off the coast of South Carolina during a September hurricane in 1857. The ship was carrying passengers and a large amount of gold bars and coins from San Francisco to New York. Figure 2.1 shows the probability map prepared for

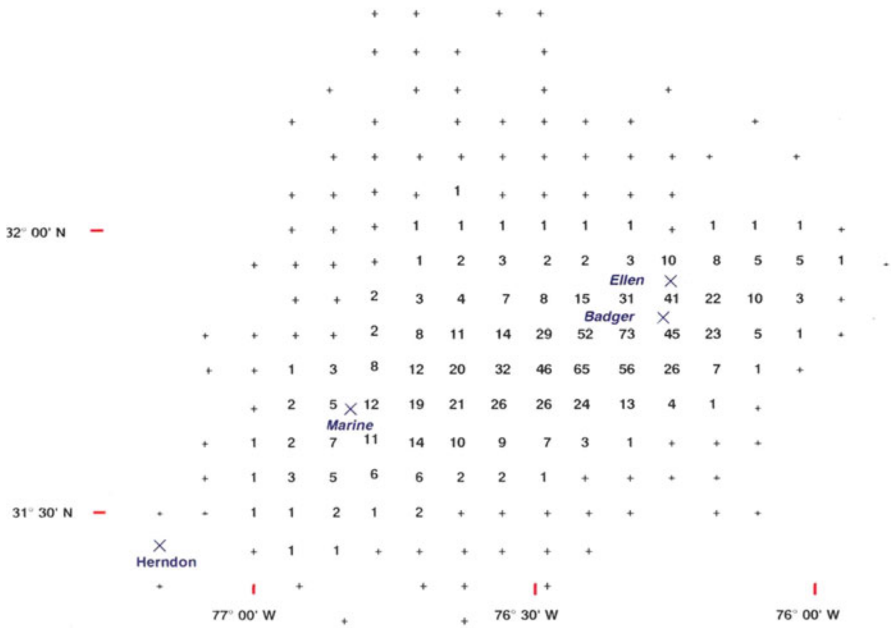


Fig. 2.1 Probability map for the location of the *SS Central America.* The probabilities are multiplied by 1000. Pluses indicate cells with probabilities between 0 and 0.001. The *Ellen* and *Marine* are ships that were in the vicinity of the *Central America* when it sank. Herndon was the Captain of the *Central America* and Badger a passenger. They both provided estimates for the location where the ship sank

the successful search for this wreck. Stone (1992) developed this map by using a Monte Carlo simulation with 12,000 points to estimate the location of the wreck. The ocean bottom was gridded into 4 nmi by 4 nmi cells. The simulation was based on historical information on the location of the wreck in ships, logs, newspaper reports, and accounts of surviving passengers. The number of points in each cell was divided by 12,000 to produce the probability in that cell.

The U. S. Coast Guard uses the Search and Rescue Optimal Planning System (Kratzke et al. (2010)) to plan maritime searches for people and ships missing at sea. The probability maps produced by this system are gridded for display purposes with the cells color-coded to represent the probabilities in the grid cells.

2.1.2 Continuous Prior Distributions

Prior distributions on continuous spaces such as those on regions of two or three-dimensional space are specified in terms of a probability density function. When the search space S is the plane, two standard priors are the uniform and the Gaussian distributions.

Uniform Prior The uniform prior distribution over a region $\mathcal{R}$ in the plane with area A is given by

$$p_u(x) = \begin{cases} 1/A \text{ for } x \in \mathcal{R} \\ 0 \quad \text{otherwise.} \end{cases} \tag{2.1}$$

Gaussian Many times a search begins with an uncertain estimate of the target's last known location based on sensor report such as a radar measurement or a visual sighting. Experience and testing has shown that errors in the measurements from many sensors have a Gaussian or normal distribution. As a result, the uncertainty in the last known location is often modeled by a bivariate normal probability distribution.

For convenience we use the (x_1, x_2) coordinate system with $(0, 0)$ at the mean of distribution and oriented so that the distribution of the target's x_1 coordinate is independent of its x_2 coordinate. Let σ_1 and σ_2 be the standard deviations of the distributions on these two coordinates. The prior density function for the target's location is

$$p_G(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left[-\frac{1}{2}\left(\frac{x_1^2}{\sigma_1^2} + \frac{x_2^2}{\sigma_2^2}\right)\right] \text{ for } \ (x_1, x_2) \in S. \tag{2.2}$$

Figure 2.2 shows a plot of this probability density function for the special case where $\sigma_1 = \sigma_2$. This is called a circular normal density function.

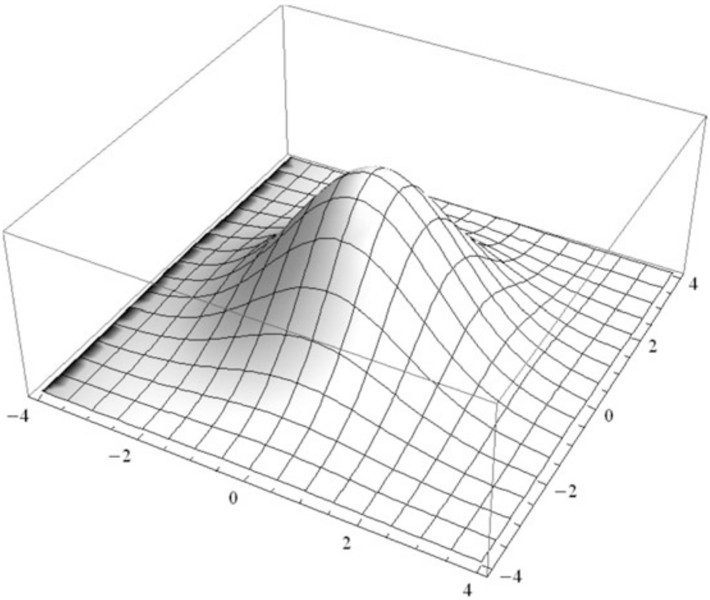


Fig. 2.2 Circular normal density function. Distances are in units of $\sigma_1 = \sigma_2$

2.2 Detection Models

To allocate search effort optimally, we must have a model for the probability of detection as a function of effort. In general one has to consider the possibility of false alarms, but for this discussion we assume that our sensor produces no false detections, that is, only true detections are reported. In this section we give a brief overview of some basic detection models. Detection modeling is discussed in more detail in Koopman (1946, 1956a, b, 1980) and Washburn (2014). See Stone (2007) Chap. 6 for a discussion of false alarms in search problems.

Let us begin by considering search in a rectangular region of area A. Suppose that we are searching with a sensor that has detection range R and that the sensor has perfect detection capability within this range. If the sensor passes within range R of the target, it will detect the target with probability 1. This is called a definite range law of detection. In this case,

$$W = 2R \tag{2.3}$$

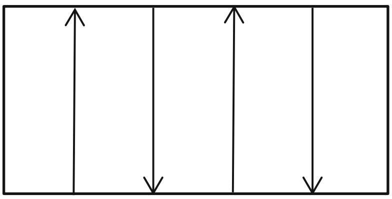
is the sweep width of the sensor. It is the width swept "clean" by the sensor as it moves through the search region. If the sensor searches over a path of length l in the rectangle, then search effort (swept area) is lW, and the search effort density (or coverage) is lW/A.

2.2.1 Detection Functions

Suppose the sensor performs a parallel track search as shown in Fig. 2.3 with paths spaced exactly W distance apart. Then the probability of detecting the target as a function of search effort density given the target is in the rectangular region is shown by the straight line in Fig. 2.4. Once the effort density reaches one, no further increase in detection probability is possible.

Often it is not possible to place the search tracks exactly one sweep width apart so that unintended gaps and overlaps occur. This will degrade the detection probability. In the limit, as our ability to place the effort where desired degrades, we come to a situation where each increment of search effort is placed according to a uniform distribution over the rectangle and is independent of where the previous increments have been placed. However, we still assume that all effort falls within the rectangle. This limiting case yields the exponential detection function which is shown in Fig. 2.4 and derived below. This function is also called the random search function

Fig. 2.3 Parallel track search



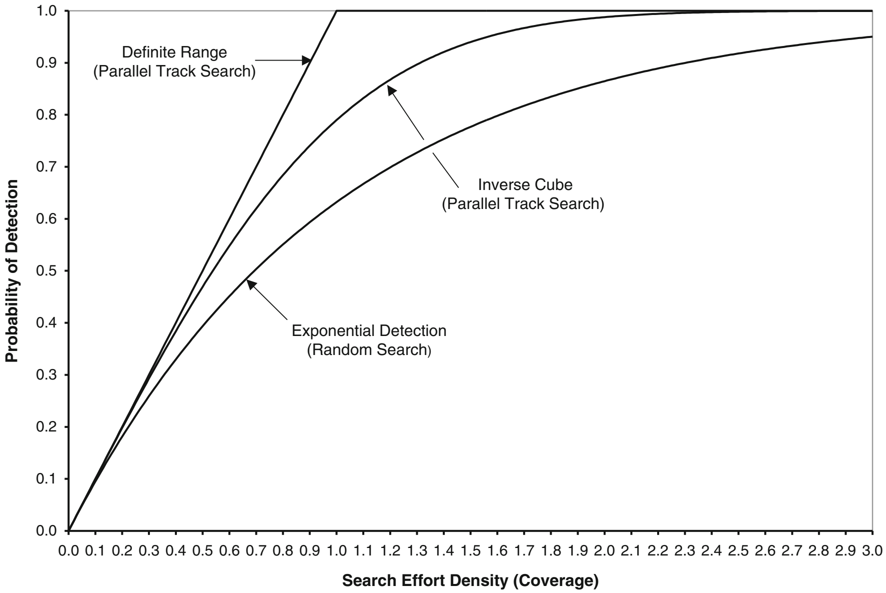


Fig. 2.4 Detection functions

or formula. Generally, the effectiveness of a parallel track search falls between the definite range and exponential detection functions. This will be the case for the inverse cube detection function derived in Sect. 2.2.2

Derivation of the Exponential Detection Function The derivation of the exponential detection function was first given by Koopman (1946, 1956b). The mathematical assumptions yielding the exponential detection function are the following.

- The location of each small increment of search effort follows a uniform distribution over the rectangle, and disjoint increments are distributed independently.
- No effort falls outside the rectangle.

Let $b(l)$ be the detection probability resulting from a search of track length l. Let h be a small increment of search effort (track length). Then

$$(1 - b(l))\,\frac{hW}{A}$$

is the probability of failing to detect the target with effort l but succeeding on the next increment h. Thus

$$b\,(l + h) = b(l) + (1 - b(l))\,\tfrac{hW}{A} \quad \text{and}$$
$$b'(l) = \lim_{h \to 0} \tfrac{b(l+h) - b(l)}{h} = (1 - b(l))\,\tfrac{W}{A}$$

Since $b(0) = 0$, the above differential equation has the well-known solution

$$b(l) = 1 - \exp\left(-lW/A\right) \quad \text{for} \quad l \geq 0. \tag{2.4}$$

In terms of the search density $z = lW/A$ and (2.4) becomes

$$b(z) = 1 - e^{-z} \quad \text{for} \quad z \geq 0 \tag{2.5}$$

which is the exponential detection function in Fig. 2.4.

Although the assumptions under which the exponential detection function is proved are somewhat artificial, it provides a useful lower bound on the effectiveness of a search in which we apply search effort uniformly over a rectangle. This is illustrated in Fig. 2.5 where we have plotted the fraction of the rectangle covered by a parallel path search with a sensor with a definite range law of detection and sweep width $W = 2R$ as in (2.3). There are 100 paths, and the intended spacing of the paths is equal to W, but we assume that the placement of each path has an independent Gaussian error with mean 0 and standard deviation σ. We performed Monte Carlo trials to compute the fraction of the rectangle covered by one or more of the paths. The dark line shows the expected fraction of the rectangle covered while the lighter lines indicate the standard deviation of the results from the mean. One can see that as σ/W reaches 5, the expected fraction of the rectangle covered becomes close to $1 - e^{-1} = 0.63$ as predicted by (2.5).

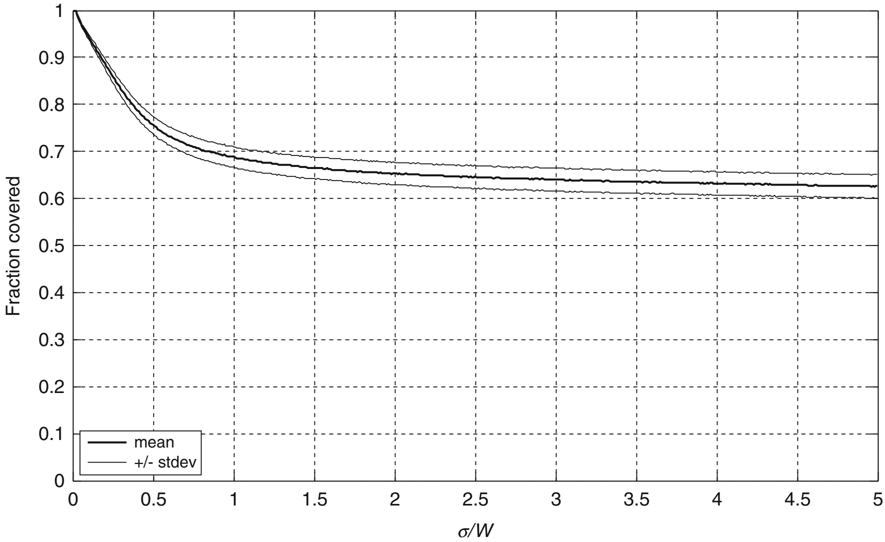
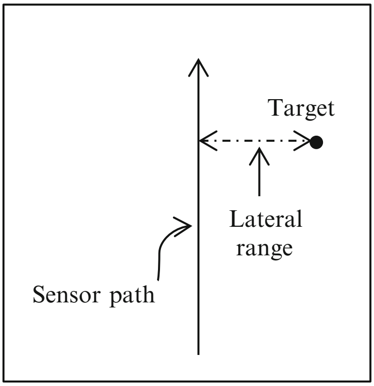


Fig. 2.5 Effectiveness of parallel path search as a function of relative navigation error, σ/W

Fig. 2.6 Lateral range



2.2.2 Lateral Range Functions

To define the lateral range function, we imagine a sensor passing a target while following a long, straight-line path as shown in Fig. 2.6. The range r at the point of closest approach of the track to the target is the lateral range. Lateral ranges are signed so that if the target is to the right of sensor as in the figure, the lateral range is positive. Let $l_d(r)$ be the probability of detecting the target when the sensor passes at lateral range r. Using the lateral range curve, we extend the definition of sweep width given in (2.3) to be

$$W = \int_{-\infty}^{\infty} l_d(r)dr. \tag{2.6}$$

A sensor with a definite range law has the following lateral range function.

$$l_d(r) = \begin{cases} 1 \text{ for } -R \leq r \leq R \\ 0 \text{ otherwise} \end{cases}$$

and sweep width $W = 2R$ as in (2.3).

One can show that the proof of the exponential detection function applies to sensors with the more general definition of sweep width given in (2.6).

Inverse Cube Lateral Range Function The inverse cube lateral range function was developed by Koopman (1946) and his colleagues to model visual detection of ship wakes. It is used by the U. S. Coast Guard, IAMSAR (2008), as one of its standard detection models for visual search. Koopman (1946, 1956b) derives this lateral range function under the following assumptions; see also Washburn (2014).

- The observer is at height h above the target which is on the ocean surface.
- The observer detects the target by seeing its wake.
- The instantaneous probability of detection γ is proportional to the solid angle $\alpha\beta$ subtended by the wake from the observer as shown in Fig. 2.7.

The solid angle subtended by a wake represented by the rectangle with sides of length a and b is the product of the angles α and β where the angles are measured in radians. Looking at the left-hand side of Fig. 2.7, we see that $\alpha = c/s$ and by similar triangles that $c/a = h/s$ so that $\alpha = ah/s^2$. Since $\beta = b/s$, we see that the solid angle is

$$\alpha\beta = \frac{abh}{s^3} = \frac{Ah}{s^3} = \frac{Ah}{(h^2 + r^2)^{3/2}} \tag{2.7}$$

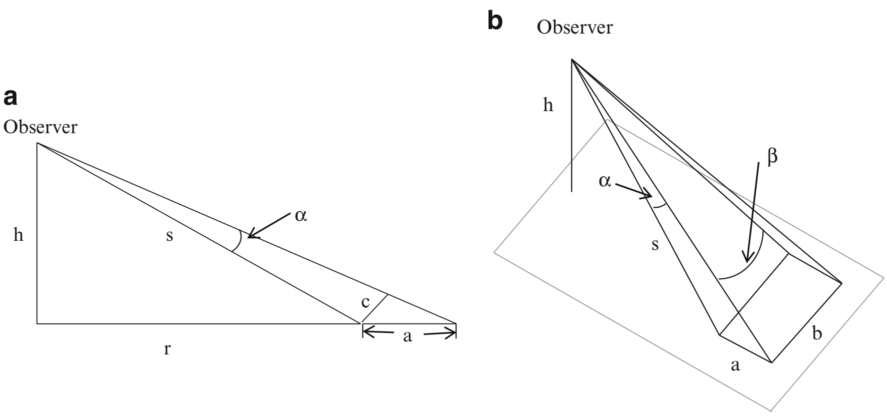


Fig. 2.7 Solid angle subtended by a wake of width a and length b

where $A = ab$ is the area of the rectangle. Since γ is proportional to the solid angle, we have for some constant k

$$\gamma = \frac{kAh}{(h^2 + r^2)^{3/2}},$$

and when r is much larger than h

$$\gamma \approx \frac{kAh}{r^3}. \tag{2.8}$$

The constant k is determined by considerations such as sea state, visibility, and type of target.

Let $\gamma(t)$ be the instantaneous detection rate at time t so that $\gamma(t)dt$ is the detection probability in a small increment of time dt. We further suppose that detection is independent over disjoint time intervals. Let $p(t_1, t_2)$ be the probability of detection over time $[t_1, t_2]$ and $q(t_1, t_2) = 1 - p(t_1, t_2)$. Then

$$q(t_1, t + dt) = q(t_1, t)\,(1 - \gamma(t)dt)\,,$$

so that

$$\frac{dq(t_1, t)}{dt} = -\gamma(t)q(t_1, t)\,.$$

Solving this differential equation, we obtain,

$$q(t_1, t) = \exp\left(-\int_{t_1}^{t} \gamma(t)dt\right) \text{ and } p(t_1, t) = 1 - \exp\left(-\int_{t_1}^{t} \gamma(t)dt\right). \tag{2.9}$$

Suppose the observer passes the target at lateral range r_0. For simplicity we choose an (x, y) coordinate system so that the observer's path goes from $-\infty$ to ∞ on the y-axis at speed v and the target is located at $(r_0, 0)$. Let the observer's position at time t be $(0, vt)$. From (2.8) we have

$$\int_{-\infty}^{\infty} \gamma(t)dt = \int_{-\infty}^{\infty} \frac{kAh}{\left(r_0^2 + (vt)^2\right)^{3/2}} dt = \frac{2kAh}{vr_0^2}. \tag{2.10}$$

So the probability of detecting the target at least once on a track which passes at lateral range r_0 is

$$l_d(r_0) = 1 - \exp\left(\frac{-2kAh}{vr_0^2}\right). \tag{2.11}$$

This is the “inverse cube” lateral range function. The inverse cube name refers to the detection rate which is proportional to the inverse cube of the range of the target from the sensor.

Inverse Cube Sweep Width From (2.11) we can compute the sweep width W for an inverse cube lateral range function by

$$W = \int_{-\infty}^{\infty} l_d(r_0)\,dr_0 = \int_{-\infty}^{\infty} \left(1 - \exp\left(\frac{-2kAh}{vr_0^2}\right)\right) dr_0 = 2\sqrt{2\pi kAh/v}. \quad (2.12)$$

See Koopman (1956b).

Inverse Cube Detection Function Let us return to the rectangular search region shown in Fig. 2.3 and consider a large number of long parallel search tracks that are spaced a distance D apart. Again let’s choose the coordinate system so that the tracks are parallel to the y-axis. The probability of detecting a target on a single track depends only on the x coordinate of the target’s location and the value of the x coordinate of the track. Let iD be the x coordinate of the ith track so that the lateral range of the track to the a target at x is $r_0 = x - iD$. Assume that detection on one track is independent of that on any other. From (2.11) and (2.12) we have the probability of failing to detect a target at x on track i is

$$\exp\left(\frac{-2kAh}{vr_0^2}\right) = \exp\left(\frac{-W^2}{4\pi(x - iD)^2}\right)$$

and the probability of failing to detect on all tracks is

$$\exp\left(-\sum_{i=-\infty}^{\infty} \frac{W^2}{4\pi(x - iD)^2}\right) = \exp\left(-\frac{\pi}{4}\left(\frac{W}{D}\right)^2 \csc\left(\frac{\pi x}{D}\right)\right) \quad (2.13)$$

where the value of the infinite sum is found by the method described in Koopman (1956b). If the distribution on the target’s location is uniform over the rectangle, the probability of detection is

$$b(W/D) = 1 - \frac{1}{D}\int_0^D \exp\left(-\frac{\pi}{4}\left(\frac{W}{D}\right)^2 \csc\left(\frac{\pi x}{D}\right)\right) dD = 2\Phi\left(\sqrt{\frac{\pi}{2}}\frac{W}{D}\right) - 1 \quad (2.14)$$

where Φ is the cumulative distribution function for a Gaussian distribution with mean 0 and the variance 1; see Koopman (1956b). Taking $z = W/D$ as search density, we obtain the inverse cube detection function in Fig. 2.4 which falls between the definite-range and exponential detection function.

When the Coast Guard uses this detection function, the search planner refers to a set of tables (see Coast Guard (2010, Appendix H)) which give sweep width for visual search as a function of the type of search object, visibility, sea state, altitude, and speed of the searching aircraft. The choice of sweep width implicitly determines the value of k in (2.12).

2.3 Optimal Search for a Stationary Target

The basic problem of optimal search for a stationary target has the following elements.

- A prior distribution on a discrete or continuous search space
- A detection function relating search effort (density) to probability of detection
- A function that specifies the allocation of search effort (search plan)
- A goal of finding the allocation that maximizes the probability of detecting the target subject to a constraint on cost.

2.3.1 Discrete Search Space: Continuous Effort

In this section we derive the solution to the optimal stationary target search problem for discrete state space and continuous effort. We begin by defining the elements of this problem in a mathematical fashion.

Prior Distribution The prior distribution is given by $p(j)$, the probability that the target is cell j for $j = 1, \ldots, J$ with

$$\sum_{j=1}^{J} p(j) = 1. \tag{2.15}$$

For convenience we assume $p(j) > 0$ for $j = 1, \ldots, J$.

Detection Function For each cell j, there is a detection function

$$b\,(j, z) = \Pr\{\text{detecting target with effort } z \mid \text{target in cell } j\} \quad \text{for} \quad z \geq 0.$$

Let $b'(j, z)$ be the derivative of $b(j, z)$ with respect to z. We assume that $b'(j, z)$ is a positive, continuous, and strictly decreasing function of z with $b'\,(j, z) < \infty$. We call this a *decreasing-rate detection function*.

Cost Function The cost of applying z effort in cell j is $c(j)z$ where $c(j) > 0$ for $j = 1, \ldots, J$. Frequently, $c(j) = 1$ in which case the cost constraint is on effort.

Rate of Return Function For $j = 1, \ldots, J$, define the rate of return function as

$$\rho\,(j, z) = b'\,(j, z)\,p(j)/c(j) \text{ for } z \geq 0. \tag{2.16}$$

Since $b'(j,z)$ is a positive, continuous, and strictly decreasing function of z, the rate of return function shares these properties. A decreasing rate of return function means that each new increment of search effort applied to the cell j produces a smaller ratio of increase in detection probability to increase in cost. In economics, this is called a decreasing rate of return. The decreasing rate of return property is common to most detection functions, e.g., the exponential detection function.

Allocation Function A search plan is an allocation function $f(j)$, $j = 1, \ldots, J$, where $f(j)$ is the amount of search effort allocated to cell j. We require that $f(j) \geq 0$ for all j. Let F be the set of allocation functions.

For an allocation f we compute the probability of detection by

$$P(f) = \sum_{j=1}^{J} b\,(j, f(j))\,p(j) \tag{2.17}$$

and the cost by

$$C(f) = \sum_{j=1}^{J} c(j) f(j). \tag{2.18}$$

Optimal Allocation Let $K > 0$ be a constraint on cost. Then $f^* \in F$ *is optimal for cost* K if

$$C\left(f^*\right) \leq K \text{ and } P\left(f^*\right) \geq P(f) \text{ for all } f \in F \text{ for which } \mathrm{C}(f) \leq K. \tag{2.19}$$

2.3.1.1 Discrete-Space Lagrangian Optimization

The concept of rate of return is crucial for finding optimal search plans. In this section we show how to find optimal plans for decreasing-rate detection functions by allocating effort so that the rate of return for the next small increment of effort is equal to a common value across all cells receiving search effort and equal to or lower than that value for cells receiving no effort. We will designate this rate of return by λ.

Fix a rate of return $\lambda > 0$. For each cell j, find $f_\lambda(j) \geq 0$ such that $\rho\,(j, f_\lambda(j)) = \lambda$ if that is possible. If this not possible, then $\rho\,(j, 0) < \lambda$, and we set $f_\lambda(j) = 0$. In summary, f_λ satisfies the following conditions.

$$\begin{aligned} \rho\,(j, f_\lambda(j)) = b'\,(j, f_\lambda(j))\,p(j)/c(j) &= \lambda \text{ if } f_\lambda(j) > 0 \\ &\leq \lambda \text{ if } f_\lambda(j) = 0. \end{aligned} \tag{2.20}$$

Because of the decreasing-rate property of b, we can show that f_λ is an optimal search allocation for cost $C(f_\lambda)$. We will do this below. By choosing different values of λ we can generate optimal plans for different costs. If we want to find an optimal plan for cost K, we need to find a λ such that $C\,(f_\lambda) = K$. If b is a decreasing-rate detection function, then $C(f_\lambda)$ will be a decreasing function of λ which means one can easily find the desired value of λ by a numerical one-dimensional search.

In the following discussion we develop an algorithm for finding optimal search plans for any desired cost and provide an example where we can solve for the optimal allocation function explicitly. We first show that f_λ is optimal for cost $C(f_\lambda)$. The proof of optimality depends on a Lagrange multiplier argument, where the rate of return λ is used as the Lagrange multiplier.

Define the *discrete-space pointwise Lagrangian* l by

$$l\,(j, z, \lambda) = b\,(j, z)\,p(j) - \lambda c(j)z \quad \text{for} \quad j = 1, \ldots, J, \; z \geq 0, \; \text{and}\; \lambda > 0. \tag{2.21}$$

Then

$$l'\,(j, z, \lambda) = b'\,(j, z)\,p(j) - \lambda c(j) \tag{2.22}$$

is the derivative of $l(j, z, \lambda)$ with respect to z.

Theorem 2.1: Lagrangian Optimization Theorem for Discrete Space and Continuous Effort Suppose the allocation $f^* \in F$ maximizes the pointwise Lagrangian for some $\lambda > 0$, i.e., for all $1 \leq j \leq J$

$$l\left(j, f^*(j), \lambda\right) \geq l\,(j, z, \lambda) \;\; \text{for}\; z \geq 0. \tag{2.23}$$

Then f^* is optimal for cost $C(f^*)$.

Proof Let f be any allocation in F such that $C(f) \leq C\,(f^*)$. By (2.23)

$$l\left(j, f^*(j), \lambda\right) \geq l\,(j, f(j), \lambda) \quad \text{for} \;\; 1 \leq j \leq J,$$

and it follows that

$$\begin{aligned}
&\sum\nolimits_{j=1}^{J} l\left(j, f^*(j), \lambda\right) \geq \sum\nolimits_{j=1}^{J} l\,(j, f(j), \lambda) \\
&P\,(f^*) - \lambda C\,(f^*) \geq P(f) - \lambda C(f) \\
&P\,(f^*) - P(f) \geq \lambda\,[C\,(f^*) - C(f)] \geq 0.
\end{aligned}$$

Thus f^* is optimal for cost $C(f^*)$. This proves the theorem.

Corollary 2.1 *Suppose b is a deceasing-rate detection function. If for some $\lambda > 0$, the allocation $f^* \in F$ satisfies*

$$\begin{aligned}
l'\left(j, f^*(j), \lambda\right) &= 0 \;\text{for}\; f^*(j) > 0 \\
&\leq 0 \;\text{for}\; f^*(j) = 0 \;\; \text{for}\; j = 1, \ldots, J,
\end{aligned} \tag{2.24}$$

then f^* is optimal for cost $C(f^*)$.

Proof Since b is a decreasing-rate detection function, $l(j, z, \lambda)$ is a concave function of z for $z \geq 0$. If $f^*(j) > 0$, then by (2.24), the derivative $l'(j, f^*(j), \lambda)$ is 0 at an interior point of the interval $[0, \infty)$, and this point is the maximum of $l(j, z, \lambda)$ for $z \geq 0$. If $f^*(j) = 0$, then since b' is decreasing it follows from (2.24) that $l'(j, z, \lambda) \leq 0$ for $z \geq 0$ and the maximum of the Lagrangian occurs at the end point $z = 0$. Thus f^* maximizes the pointwise Lagrangian and is optimal for cost $C(f^*)$.

Bounded Allocation Functions There may be times when there is an upper bound $B > 0$ on the search effort that can be placed in a cell. In this case we can employ the discrete-space Lagrangian optimization in modified form. Let F_B be the set of allocation functions f such that

$$0 \leq f(j) \leq B \text{ for } j = 1, \ldots, J.$$

Theorem 2.1′: Lagrangian Optimization Theorem for Discrete Space and Bounded Continuous Effort Suppose the allocation $f^* \in F_B$ maximizes the pointwise Lagrangian for some $\lambda > 0$, i.e., for all $1 \leq j \leq J$

$$l\left(j, f^*(j), \lambda\right) \geq l(j, z, \lambda) \text{ for } 0 \leq z \leq B. \tag{2.25}$$

Then f^* is optimal for cost $C(f^*)$ over $f \in F_B$.

Corollary 2.1′ *Suppose b is a deceasing-rate detection function. If for some $\lambda > 0$, the allocation $f^* \in F_B$ satisfies*

$$\begin{aligned} l'(j, f^*(j), \lambda) &\geq 0 \text{ for } f^*(j) = B \\ &= 0 \text{ for } 0 < f^*(j) < B \\ &\leq 0 \text{ for } f^*(j) = 0 \qquad \text{for } j = 1, \ldots, J, \end{aligned} \tag{2.26}$$

then f^* is optimal for cost $C(f^*)$.

The proofs of Theorem 2.1′ and Corollary 2.1′ are straight-forward extensions of the proofs of Theorem 2.1 and Corollary 2.1.

The Lagrangian Optimization Theorem given above and the continuous space form given in Sect. 2.3.3 are due to Everett (1963). Chapter II of Stone (2007) shows that conditions (2.23) are also necessary for optimality for decreasing-rate detection functions.

2.3.1.2 Optimal Plan: Discrete Space, Continuous Effort

We now present an algorithm for computing the optimal search plan for discrete-space, continuous-effort search with a decreasing-rate detection function. Since the detection function is decreasing-rate, it is concave and one can use standard constrained optimization routines such as the ones provided by MATLAB to find optimal plans. We present the method below because it provides insight into the nature of the optimal plan and generalizes easily to a continuous search space.

If b is a decreasing-rate detection function, then $\rho(j,\cdot)$ is a continuous, strictly decreasing function. Observe that,

$$\rho(j,\infty) \equiv \lim_{z\to\infty} \rho(j,z) = 0 \text{ for } 1 \le j \le J.$$

As a result, the inverse function $\rho^{-1}(j,\cdot)$ exists and is defined on $(0,\rho(j,0)]$. It satisfies

$$\rho\left(j,\rho^{-1}(j,\lambda)\right) = \lambda \text{ for } 0 < \lambda \le \rho(j,0) \text{ and } 1 \le j \le J. \tag{2.27}$$

For convenience define $\rho^{-1}(j,\lambda) = 0$ for $\lambda > \rho(j,0)$. Now let

$$f_\lambda(j) = \rho^{-1}(j,\lambda) \text{ for } 0 < \lambda < \infty \text{ and } 1 \le j \le J. \tag{2.28}$$

Then f_λ satisfies

$$\begin{aligned} l'(j,f_\lambda(j),\lambda) &= 0 \text{ for } f_\lambda(j) > 0 \\ &\le 0 \text{ for } f_\lambda(j) = 0, \end{aligned}$$

and by Corollary 2.1, f_λ is optimal for cost $C(f_\lambda)$.

We can now generate optimal plans by the use of (2.28). To do this we define

$$\mathbf{K}(\lambda) = \sum\nolimits_{j=1}^{J} c(j)\rho^{-1}(j,\lambda) \text{ for } \lambda > 0. \tag{2.29}$$

One can verify that $\mathbf{K}$ is a continuous function of λ with following properties

$$\begin{aligned} &\mathbf{K}(\lambda) = 0 \text{ for } \lambda \ge \lambda_{\max} \equiv \max_j \rho(j,0) \\ &\lim_{\lambda\to 0} \mathbf{K}(\lambda) = \infty. \end{aligned}$$

Furthermore $\mathbf{K}$ is strictly decreasing from ∞ to 0 as λ increases from 0 to $\lambda_{\max}$, so the inverse function $\mathbf{K}^{-1}$ exists and is defined on $(0,\infty)$. It satisfies

$$\mathbf{K}^{-1}(K) = \lambda \text{ such that } C(f_\lambda) = K \text{ for } \lambda > 0. \tag{2.30}$$

Thus, for any cost $K > 0$, we may set $\lambda = \mathbf{K}^{-1}(K)$ and obtain a plan f_λ from (2.28) that is optimal for cost K.

Bounded Allocation Functions If the allocation functions are bounded by $B > 0$, then we can use the method described above to find an optimal plan $f^* \in F_B$ for cost K by modifying ρ^{-1} as follows. Define

$$\rho_B^{-1}(j,\lambda) = \min\left\{B,\rho^{-1}(j,\lambda)\right\} \text{ for } 0 < \lambda \le \rho(j,0).$$

Clearly $\rho^{-1} = \rho_B^{-1}$ when $B = \infty$. With this in mind, we can replace ρ^{-1} by ρ_B^{-1} in (2.29) and solve for the optimal plan F_B with this modified definition of $\mathbf{K}$. If $B < \infty$, then there will be a maximum amount of effort that can be allocated to the search.

Usually one has to compute $\mathbf{K}$ numerically to find the λ that produces $\mathbf{K}(\lambda) = K$ and obtain f_λ in (2.32). Section 5.2 of Washburn (2014) gives an algorithm for finding λ in a finite number of steps when the detection function is exponential and $B = \infty$. Since $\mathbf{K}$ is decreasing, a simple one-dimensional search can be used to find the necessary λ.

Optimal Allocation for Cost *K* We now summarize the method described above for finding optimal allocations.

Algorithm
Optimal Search Plan for Cost $K \le JB$
Discrete Space, Continuous Effort

Assume b is a decreasing-rate detection function. Let $0 < B \le \infty$.

Define

$$\begin{aligned} &\rho(j,z) = b'(j,z)\,p(j)/c(j) \text{ for } 0 \le z \le B \text{ and } 1 \le j \le J \\ &\mathbf{K}(\lambda) = \sum_{j=1}^{J} c(j)\rho_B^{-1}(j,\lambda) \text{ for } \lambda > 0. \end{aligned} \tag{2.31}$$

Compute

$$\begin{aligned} &\lambda = \mathbf{K}^{-1}(K) \text{ and} \\ &f_\lambda(j) = \rho^{-1}(j,\lambda) \text{ for } 0 < \lambda < \infty \text{ and } 1 \le j \le J. \end{aligned} \tag{2.32}$$

Then f_λ is optimal within F_B for cost K.

2.3.1.3 Optimal Plan: Discrete Space, Exponential Detection Function

In this example, we are able to compute the optimal plan analytically. Suppose $A_j > 0$ is the area of cell j and $W_j > 0$ is the sweep width of the sensor in cell j. Let

$$b(j,z) = 1 - \exp\left(-W_j z/A_j\right) \quad \text{for } z \ge 0,\ 1 \le j \le J.$$

be the detection function where z is measured in units of track length. For convenience of notation, let

$$\alpha_j = W_j/A_j \ \text{ for } \ 1 \le j \le J.$$

The ratio α_j may be thought of as an efficiency coefficient. Search is more efficient or effective in cells with higher values of α_j. We have

$$\begin{aligned}\rho(j,z) &= \tfrac{p(j)}{c(j)}\alpha_j \exp\left(-\alpha_j z\right) \text{ and} \\ \rho^{-1}(j,\lambda) &= -\tfrac{1}{\alpha_j}\left[\ln\left(\tfrac{c(j)}{\alpha_j p(j)}\lambda\right)\right]^+\end{aligned} \tag{2.33}$$

where

$$a^+ = \begin{cases} a \text{ if } a \geq 0 \\ 0 \text{ otherwise.}\end{cases}$$

Order the cells so that

$$\frac{p(1)}{c(1)}\alpha_1 \geq \frac{p(2)}{c(2)}\alpha_2 \geq \cdots \geq \frac{p(J)}{c(J)}\alpha_J. \tag{2.34}$$

If one thinks about the optimal search developing in time, then search begins in cell 1, which has the highest rate of return, and continues solely in that cell until it reaches the effort z where

$$\frac{p(1)}{c(1)}\alpha_1 \exp\left(-\alpha_1 z\right) = \frac{p(2)}{c(2)}\alpha_2, \tag{2.35}$$

i.e., when

$$z = \frac{1}{\alpha_1}\left[\ln\left(\alpha_1 p(1)c(2)\right) - \ln\left(\alpha_2 p(2)c(1)\right)\right]. \tag{2.36}$$

Search then expands into cell 2 so that additional effort z_1 and z_2 in cells 1 and 2 satisfies

$$\left[\frac{p(1)}{c(1)}\alpha_1 \exp\left(-\alpha_1 z\right)\right]e^{-\alpha_1 z_1} = \left[\frac{p(2)}{c(2)}\alpha_2\right]e^{-\alpha_2 z_2} \tag{2.37}$$

to maintain the equality of the rates of return. This implies $\alpha_1 z_1 = \alpha_2 z_2$. Search will continue this way until it expands into the third cell after which the incremental effort will be added to each cell in inverse proportion to α_j for $j = 1, 2, 3$. If the search is unsuccessful, this process will continue until cell j is receiving increments of effort that are proportional to $1/\alpha_j$ for $j = 1, \ldots, J$.

Let

$$y_j = \ln\left(\alpha_j p(j) c(j+1)\right) - \ln\left(\alpha_{j+1} p(j+1) c(j)\right) \text{ for } 1 \leq j \leq J-1. \tag{2.38}$$

From the discussion above we can see that y_j/α_j is the amount of effort placed in cell j before the search expands to cell $j+1$ and that

$$\frac{1}{\alpha_j}\sum\nolimits_{k=j}^{i} y_k \text{ for } 1 \leq j \leq i$$

is the total effort allocated to cell j before the search expands to cell $i+1$. From this we calculate that

$$\sum\nolimits_{j=1}^{i} \frac{1}{\alpha_j} \sum\nolimits_{k=j}^{i} y_k$$

is the total effort place in all cells before search expands to cell $i + 1$. The cost of this effort is

$$G(i) = \sum\nolimits_{j=1}^{i} \frac{c(j)}{\alpha_j} \sum\nolimits_{k=j}^{i} y_k \text{ for } 1 \leq i \leq J-1. \tag{2.39}$$

To find an allocation f^* that is optimal for cost $0 < K \leq G(J-1)$, set $G(0) = 0$ and find the value of i for which $G(i-1) < K \leq G(i)$. Set

$$a = \frac{K - G(i-1)}{G(i) - G(i-1)}.$$

Then

$$f^*(j) = \begin{cases} \frac{1}{\alpha_j} \sum\nolimits_{k=j}^{i-1} y_k + \dfrac{a}{\alpha_j} y_i & \text{for } 1 \leq j \leq i \\ 0 & \text{for } j > i. \end{cases} \tag{2.40}$$

is optimal for cost K. In (2.40) we follow the convention that the sum over an empty set of indices is 0.

If $K > G(J-1)$, set

$$a = (K - G(J-1)) / \sum\nolimits_{j=1}^{J} c(j)/\alpha_j.$$

Then f^* defined below is optimal for cost K. Specifically,

$$f^*(j) = \frac{1}{\alpha_j} \sum\nolimits_{k=j}^{J-1} y_k + \frac{a}{\alpha_j} \text{ for } 1 \leq j \leq J. \tag{2.41}$$

We summarize this method of finding optimal allocations in the following algorithm description.

Algorithm
Optimal Search Plan for Cost K
Discrete Space, Exponential Detection Function, $B = \infty$

Let $\alpha_j = W_j/A_j$, and order the cells so that

$$\frac{p(1)}{c(1)}\alpha_1 \geq \frac{p(2)}{c(2)}\alpha_2 \geq \cdots \geq \frac{p(J)}{c(J)}\alpha_J. \tag{2.42}$$

(continued)

Compute

$$y_j = \ln\left(\alpha_j p(j) c\,(j+1)\right) - \ln\left(\alpha_{j+1} p\,(j+1)\,c(j)\right) \quad \text{for } 1 \le j \le J-1.$$

Set $G(0) = 0$, and compute

$$G(i) = \sum\nolimits_{j=1}^{i} \frac{c(j)}{\alpha_j} \sum\nolimits_{k=j}^{i} y_k \quad \text{for } 1 \le i \le J-1. \tag{2.43}$$

If $G\,(i-1) < K \le G(i)$ for some $1 \le i \le J-1$, set

$$a = \frac{K - G\,(i-1)}{G(i) - G\,(i-1)} \quad \text{and}$$

$$f^*(j) = \begin{cases} \frac{1}{\alpha_j} \sum\nolimits_{k=j}^{i-1} y_k + \dfrac{a}{\alpha_j} y_i & \text{for } 1 \le j \le i \\ 0 & \text{for } j > i. \end{cases}$$

Then f^* is optimal for cost K.
If $K > G\,(J-1)$, set

$$a = \frac{K - G\,(J-1)}{\sum\nolimits_{j=1}^{J} c(j)/\alpha_j} \quad \text{and}$$

$$f^*(j) = \frac{1}{\alpha_j} \sum\nolimits_{k=j}^{J-1} y_k + \frac{a}{\alpha_j} \quad \text{for } 1 \le j \le J.$$

Then f^* is optimal for cost K.

The algorithm given above is due to Charnes and Cooper (1958).

Leveling the Posterior Distribution In the case where all the cost coefficients $c(j)$ are equal to a common value c and the efficiency parameters α_j are equal to a common value α, the cells are ordered from highest prior probability to lowest and (2.35) becomes

$$p(1) \exp(-\alpha z) = p(2). \tag{2.44}$$

The left-hand side of (2.44) is proportional to the posterior probability of the target being in cell 1 given failure to detect the target with z search effort. Correspondingly (2.38) becomes

$$y_j = \ln p(j) - \ln p\,(j+1) \quad \text{for } 1 \le j \le J-1, \tag{2.45}$$

so that y_j/α becomes the amount of effort placed in cell j before search expands to cell $j+1$. The optimal search can be described as beginning in the highest probability cell and then expanding the search to the next highest cell when the posterior probability in cell 1 equals the posterior probability in cell 2. As search continues, equal increments are added to all cells receiving search until the common posterior in these cells fall to the value in the highest unsearched cell. The search proceeds to level the posterior distribution given failure to detect in all cells receiving search. Those not receiving search have a lower posterior probability of containing the target than the ones receiving search effort.

2.3.2 Discrete Search Space and Effort

In this section we derive the solution to the optimal stationary target search problem for discrete state space and discrete search effort when the detection function is decreasing-rate. The state space is the set of J cells as in Sect. 2.3.1 with a prior probability distribution on target location specified by $p(j)$ for $j = 1, \ldots, J$.

Detection Function Effort is allocated to cells in discrete looks. There is a detection function

$$b\,(j,k) = \Pr\{\text{detecting target on or before the } k\text{th look} \mid \text{target in cell } j\}$$
$$\text{for } k = 0, 1, \ldots \text{ and } 1 \le j \le J.$$

We assume $b\,(j,0) = 0$. Let

$$b'\,(j,k) = b\,(j,k) - b\,(j,k-1) \quad \text{for} \quad k \ge 1 \text{ and } 1 \le j \le J.$$

We use the notation b' here and l' below to indicate the discrete counterpart of a derivative.

If $b'(j,k)$ is a strictly decreasing function of k for $1 \le j \le J$, then we call b a *decreasing-rate detection function*. As an example, the detection function defined in Sect. 2.3.2.2 has a strictly decreasing rate of return.

Cost Function The cost of applying k looks in cell j is $c(j)k$ where $c(j) > 0$ for $0 \le 1 \le J$.

Rate of Return Function For $j = 1, \ldots, J$, define the rate of return function as

$$\rho\,(j,k) = b'\,(j,k)\,p(j)/c(j) \text{ for k} \ge 1. \tag{2.46}$$

Allocation Function A search plan is an allocation function $f(j)$, $j = 1, \ldots, J$ where $f(j)$ is the number of looks allocated to cell j. We require that $f(j) \ge 0$ for all j. Let F be the set of allocation functions.

For an allocation f we compute the probability of detection $P(f)$ and cost $C(f)$ as in (2.17) and (2.18).

Optimal Allocation The allocation $f^* \in F$ is *optimal for cost K* if

$$C\left(f^*\right) \leq K \text{ and } P\left(f^*\right) \geq P(f) \text{ for all } f \in F \text{ for which } \mathrm{C}(f) \leq K. \tag{2.47}$$

Pointwise Lagrangian Define the *pointwise Lagrangian l* for discrete space and effort by

$$l\,(j, k, \lambda) = b\,(j, k)\,p(j) - \lambda c(j)k \quad \text{for} \quad 1 \leq j \geq J, \;\; \mathrm{k} = 0, 1, \ldots, \;\; \text{and}\; \lambda > 0. \tag{2.48}$$

The derivative l' is defined as

$$l'\,(j, k, \lambda) = b'\,(j, k)\,p(j) - \lambda c(j). \tag{2.49}$$

Theorem 2.2. Lagrangian Optimization Theorem for Discrete Space and Effort *Suppose the allocation $f^* \in F$ maximizes the pointwise Lagrangian for some $\lambda > 0$,* i.e., *for all $1 \leq j \leq J$*

$$l\left(j, f^*(j), \lambda\right) \geq l\,(j, k, \lambda) \;\; \textit{for}\; \mathrm{k} = 0, 1, \ldots. \tag{2.50}$$

Then f^* is optimal for cost $C(f^*)$.

Proof The proof of this theorem is the same as for Theorem 2.1 in Sect. 2.3.1.1.

Corollary 2.2 *If for some $\lambda > 0$, the allocation $f^* \in F$ satisfy satisfies*

$$l'\,(j, k, \lambda) \begin{array}{l} \geq 0 \text{ for } 1 \leq k \leq f^*(j) \\ \leq 0 \text{ for } k > f^*(j) \end{array} \quad \text{for} \;\; j = 1, \ldots, J, \tag{2.51}$$

then f^* is optimal for cost $C(f^*)$.

Proof Suppose we have a $\lambda > 0$ and a search plan f^* that satisfies (2.51).

(Note that when $f^*(j) = 0$, only the second line in (2.51) applies.) Observe that

$$b\left(j, f^*(j)\right)p(j) - \lambda c(j) f^*(j) = \sum_{k=1}^{f^*(j)} \left[b'\,(j, k)\,p(j) - \lambda c(j)\right] \tag{2.52}$$

where we follow the convention that a sum from $k = 1$ to 0 is 0. Consider any allocation $f \in F$. For each j, there are two possibilities. First, suppose $f(j) > f^*(j)$. Then

$$\begin{aligned} &[b\,(j, f^*(j))\,p(j) - \lambda c(j) f^*(j)] - [b\,(j, f(j))\,p(j) - \lambda c(j) f(j)] \\ &= -\sum_{f^*(j)+1}^{f(j)} \left[b'\,(j, k)\,p(j) - \lambda c(j)\right] \geq 0 \end{aligned} \tag{2.53}$$

because by (2.51) all the terms in the sum on the right-hand side of (2.53) are less than or equal to 0. Thus

$$b\left(j,f^*(j)\right)p(j) - \lambda c(j)f^*(j) \geq b\left(j,f(j)\right)p(j) - \lambda c(j)f(j). \tag{2.54}$$

If $f(j) \leq f^*(j)$, then a similar argument shows that (2.54) holds. Thus f^* maximizes the pointwise Lagrangian, and the conditions of Theorem 2.2 hold. As a result, f^* is optimal for $C(f^*)$.

Observe that if b is a decreasing-rate detection function and f^* satisfies

$$\begin{array}{l} \rho(j,k) \geq \lambda \text{ for } 1 \leq k \leq f^*(j) \\ \leq \lambda \text{ for } k > f^*(j) \end{array} \quad \text{for } j = 1,\ldots,J, \tag{2.55}$$

then f^* satisfies (2.51) and is optimal for cost $C(f^*)$.

2.3.2.1 Optimal Plan for Discrete Space and Effort with a Decreasing-Rate Detection Function

We now present an algorithm for finding optimal plans for a decreasing-rate detection function. If we have function of two variables such as $\varphi(j,n)$, we use the notation $\varphi(\cdot,n)$ to indicate the function of one variable obtained by holding the second fixed at n.

Algorithm
Optimal Search Plan for Discrete Space and Effort

Suppose b is a decreasing-rate detection function. We construct an optimal plan $\varphi^*(\cdot,n)$ for any number of looks n in a recursive fashion as follows.

Set $\varphi^*(j,0) = 0$ for $1 \leq j \leq J$, and suppose we have found $\varphi^*(\cdot,n-1)$.

Find j_n such that

$$\rho\left(j_n,\varphi^*(j_n,n-1)+1\right) \geq \rho\left(j,\varphi^*(j,n-1)+1\right) \text{ for } 0 \leq j \leq J. \tag{2.56}$$

Set

$$\varphi^*(j,n) = \begin{cases} \varphi^*(j,n-1)+1 & \text{for } j = j_n \\ \varphi^*(j,n-1) & \text{for } j \neq j_n. \end{cases} \tag{2.57}$$

Then $\varphi^*(\cdot,n)$ will be optimal for n looks.

To see that $\varphi^*(\cdot,n)$ is optimal for n looks, we show that $f^* = \varphi^*(\cdot,n)$ satisfies (2.55) for $\lambda_n = \rho(j_n,\varphi^*(j_n,n))$.

Since b is decreasing-rate, we see from (2.56) that (2.55) holds for $j = j_n$ when $\lambda = \lambda_n$. We show by induction on n that (2.55) holds for $j \neq j_n$ when $\lambda = \lambda_n$. Clearly (2.55) holds for $n = 1$. For $n > 1$, suppose (2.55) holds for $n - 1$, and note that $\varphi^*(j,n) = \varphi^*(j,n-1)$ for $j \neq j_n$. Thus for $j \neq j_n$,

$$\rho\left(j, \varphi^*(j,n)\right) = \rho\left(j, \varphi^*(j,n-1)\right) \geq \lambda_{n-1} \geq \lambda_n \text{ for } 1 \leq k \leq \varphi^*(j,n)\,.$$

From (2.56) and the fact that b is decreasing-rate, it follows that $\rho\left(j, \varphi^*(j,n)\right) \leq \lambda_n$ for $k > \varphi^*(j,n)$ for $j \neq j_n$ and (2.55) is satisfied.

2.3.2.2 Example for Discrete Space and Effort

As in Sect. 2.3.1.2, one can numerically solve for the optimal allocation in the example below by using a standard integer optimization routine. In this section, we present an explicit method for finding optimal allocations to give the reader insight into the nature of the optimal plan and provide an alternative to using a solver.

Suppose that each look in cell j has probability q_j of detecting the target given it is in that cell for $1 \leq j \leq J$ and that each look provides an independent opportunity to detect the target. Then

$$b(j,k) = 1 - \left(1 - q_j\right)^k \text{ for } k \geq 0 \text{ and } b'(j,k) = q_j\left(1 - q_j\right)^{k-1} \text{for } k \geq 1.$$

Note that b is a decreasing-rate detection function because $b'(j,k)$ is a strictly decreasing function of k for all j. As a result, the rate of return function ρ, shown below, is likewise strictly decreasing.

$$\rho(j,k) = \frac{p(j)b'(j,k)}{c(j)} = \frac{p(j)q_j\left(1-q_j\right)^{k-1}}{c(j)}.$$

Following the algorithm above, we construct the optimal plan $\varphi^*(\cdot,n)$ for n looks recursively. Set $\varphi^*(j,0) = 0$ for $1 \leq j \leq J$, and suppose we have found $\varphi^*(\cdot,n-1)$. Obtain $\varphi^*(\cdot,n)$ by finding j_n such that

$$\frac{p\left(j_n\right) q_{j_n}\left(1-q_{j_n}\right)^{\varphi^*(j_n,n-1)}}{c\left(j_n\right)} \geq \frac{p(j)q_j\left(1-q_j\right)^{\varphi^*(j,n-1)}}{c(j)} \text{ for } 1 \leq j \leq J$$

and setting

$$\varphi^*(j,n) = \begin{cases} \varphi^*(j,n-1) + 1 & \text{for } j = j_n \\ \varphi^*(j,n-1) & \text{for } j \neq j_n. \end{cases}$$

One can think about the optimal plan as being executed one look at a time. It proceeds by placing its next look in the cell with the highest rate of return given the previous unsuccessful looks. If $c(j) = \overline{c}$ and $q_j = \overline{q}$ for all j, then the optimal plan proceeds by placing its next look in the cell with the highest posterior probability given failure of the previous looks to detect the target.

2.3.3 Continuous Search Space and Effort

In this section we derive the solution to the optimal stationary target search problem for a continuous search space S and continuous effort. We begin by defining the elements of this problem in a mathematical fashion.

Prior Distribution The search space S is continuous. It is typically a subset of the usual two or three-dimensional space. The prior distribution is given by a probability density function p defined on S. The probability that the target is in a subset $\mathcal{R}$ of S is

$$\int_{\mathcal{R}} p(x)dx \tag{2.58}$$

and the probability the target is in S is 1.

Detection Function There is a detection function b defined as follows. For $x \in S$,

$$b\,(x, z) = \Pr\{\text{detecting target with effort density } z \,|\, \text{target at } x\} \quad \text{for} \quad z \geq 0.$$

Let $b'(x, z)$ be the derivative of $b(x, z)$ with respect to z. We assume that $b'(x, z)$ is a positive, continuous, and strictly decreasing function of z. We call this a *decreasing-rate detection function*.

The notion of detection function for continuous spaces is somewhat idealized. We assume the probability of detecting a target located at x depends on the effort density in a small neighborhood of x with the dependence given by $b\,(x, \cdot)$. If effort is measured by swept area, then effort density at x is the ratio of swept area in a small neighborhood of x to the area of that neighborhood. In the idealized model, effort density is the limit of this ratio as the area of the neighborhood approaches 0.

Cost Function The cost of applying z effort density at $x \in S$ is $c(x)z$ where $c(x) > 0$ for $x \in S$.

Rate of Return Function For $x \in S$, define the rate of return function by

$$\rho\,(x, z) = b'\,(x, z)\,p(x)/c(x) \text{ for } z \geq 0. \tag{2.59}$$

Since $b'(x, z)$ is a positive, continuous, and strictly decreasing function of z, the rate of return function shares these properties.

Allocation Function A search plan is an allocation function $f(x)$ for $x \in S$ where $f(x)$ is the search effort density allocated to point x. We require that $0 \leq f(x) < \infty$ for all x. Let F be the set of allocation functions and let

$$F_B = \{f \in F : 0 \leq f(x) \leq B \text{ for } x \in S\} \quad \text{for} \quad 0 < B < \infty.$$

We define $F_\infty = F$.

For an allocation f, we compute its probability of detection and cost by

$$P(f) = \int_S b\,(x, f(x))p(x)dx \tag{2.60}$$

$$C(f) = \int_S c(x)f(x)dx. \tag{2.61}$$

2.3.3.1 Continuous-Space Lagrangian Optimization

This section presents the continuous versions of the Lagrangian optimization results in Sect. 2.3.1.1.

The Case Where $B < \infty$. If the bound $B < \infty$, define the *continuous-space pointwise Lagrangian* l by

$$l\,(x, z, \lambda) = b\,(x, z)\,p(x) - \lambda c(x)z \ \text{ for } \ x \in S, \ 0 \le z \le B, \ \text{and } \lambda > 0. \tag{2.62}$$

Define

$$l'\,(x, z, \lambda) = b'\,(x, z)\,p(x) - \lambda c(x) \tag{2.63}$$

which is the derivative of $l(x, z, \lambda)$ with respect to z.

Theorem 2.3: Lagrangian Optimization Theorem for Continuous Space Suppose the allocation $f^* \in F_B$ maximizes the pointwise Lagrangian for some $\lambda > 0$, i.e.,

$$l\left(x, f^*(j), \lambda\right) \ge l\,(x, z, \lambda) \ \text{ for } 0 \le z \le B \text{ and } x \in S \tag{2.64}$$

Then f^* is optimal in F_B for cost $C(f^*)$.

Proof The proof is analogous to the one given in Theorem 2.1 with integration over S replacing summation over j.

Corollary 2.3 *Suppose b is a deceasing-rate detection function. If for some $\lambda > 0$, the allocation $f^* \in F_B$ satisfies*

$$\begin{aligned} l'\,(x, f^*(x), \lambda) &\ge 0 \text{ for } f^*(x) = B \\ &= 0 \text{ for } 0 < f^*(x) \le B \qquad \text{for } \ x \in S, \\ &\le 0 \text{ for } f^*(x) = 0, \end{aligned} \tag{2.65}$$

then f^* is optimal for cost $C(f^*)$.

Proof The proof of this corollary is analogous to the proof of Corollary 2.1.

The Case Where $B = \infty$. If the bound $B = \infty$, then minor modifications are needed to accommodate the restriction that we can't have $z = \infty$. Specifically (2.64) and (2.65) become

$$l\left(x, f^*(j), \lambda\right) \geq l(x, z, \lambda) \text{ for } 0 \leq z < \infty \text{ and } x \in S \tag{2.65a}$$

$$\begin{aligned} l'(x, f^*(x), \lambda) &= 0 \text{ for } 0 < f^*(x) < \infty \\ &\leq 0 \text{ for } f^*(x) = 0 \end{aligned} \quad \text{for } x \in S. \tag{2.66a}$$

Measure Theory Considerations Those familiar with measure theory, will recognize that some assumptions are required to insure that the functions $f \in F_B$ are integrable. To guarantee this, we assume that all functions are Borel measurable. In practice, this places no restriction on the functions p, c, b, or f.

In addition, they will recognize that we can weaken the assumptions that (2.64) and (2.65) hold for all $x \in S$ by assuming that they hold for almost every $x \in S$, and the theorem and corollary remain true. This will be true for most of the statements that involve for $x \in S$, but we will not remark on it further.

Remark Theorem 2.1.5 in Stone (2007) shows that the satisfaction of (2.64) for almost every $x \in S$ is a necessary and sufficient condition for an optimal search plan for any Borel measureable detection function. This means that for a decreasing-rate detection function, the conditions in (2.65) are necessary for $f^* \in F_B$ to be T-optimal for cost $C(f^*)$.

2.3.3.2 Optimal Plan: Continuous Space, Continuous Effort

As in Sect. 2.3.1.2, the inverse function $\rho^{-1}(x, \cdot)$ exists and is defined on $(0, \rho(x, 0)]$. By definition

$$\rho\left(x, \rho^{-1}(x, \lambda)\right) = \lambda \text{ for } 0 < \lambda \leq \rho(x, 0)\,. \tag{2.66}$$

As before we extend the definition of $\rho^{-1}(x, \cdot)$ so that $\rho^{-1}(x, \lambda) = 0$ for $\lambda > \rho(x, 0)$ and define

$$\rho_B^{-1}(x, \lambda) = \min\left\{B, \rho_B^{-1}(x, \lambda)\right\}$$

$$f_\lambda(x) = \rho_B^{-1}(x, \lambda) \quad \text{for} \quad 0 < \lambda < \infty \quad \text{and} \quad x \in S. \tag{2.67}$$

Then f_λ satisfies

$$\begin{aligned} l'(x, f_\lambda(x), \lambda) &\geq 0 \text{ for } f_\lambda(x) = B \\ &= 0 \text{ for } 0 < f_\lambda(x) < B \\ &\leq 0 \text{ for } f_\lambda(x) = 0 \end{aligned} \tag{2.68}$$

and maximizes the continuous pointwise Lagrangian for λ. By Theorem 2.3, f_λ is optimal in F_B for cost $C(f_\lambda)$. As in Sect. 2.3.1.2, we define

$$\begin{aligned} &\mathbf{K}(\lambda) = \int_S c(x) f_\lambda(x) dx \text{ for } \lambda > 0 \text{ and} \\ &\mathbf{K}^{-1}(K) \text{ for } 0 < K < \infty \end{aligned} \tag{2.69}$$

where $\mathbf{K}^{-1}$ satisfies

$$\mathbf{K}^{-1}(K) = \lambda \text{ such that } C(f_\lambda) = K \text{ for } \lambda > 0. \tag{2.70}$$

Thus, for any cost $K > 0$, we may set $\lambda = \mathbf{K}^{-1}(K)$ and obtain a plan f_λ from (2.67) that is optimal for cost K. When $B < \infty$, there may be an finite upper bound on the cost $C(f)$ for $f \in F_B$.

Optimal Allocation for Cost K We summarize this method of finding optimal plans as follows.

Algorithm
Optimal Search Plan for Cost K
Continuous Space, Continuous Effort

Let b be a decreasing-rate detection function.
Define

$$\begin{aligned} &\rho(x, z) = b'(x, z)\, p(x)/c(x) \text{ for } z \geq 0 \text{ and } x \in S \\ &\mathbf{K}(\lambda) = \int_S c(x) \rho_B^{-1}(x, \lambda)\, dx \text{ for } \lambda > 0. \end{aligned} \tag{2.71}$$

Compute

$$\begin{aligned} &\lambda = \mathbf{K}^{-1}(K) \text{ and} \\ &f_\lambda(x) = \rho_B^{-1}(x, \lambda) \ \text{ for } x \in S. \end{aligned} \tag{2.72}$$

Then f_λ is optimal in F_B for cost K.

As noted in Sect. 2.3.1.2, usually one has to compute $\mathbf{K}$ numerically to find the λ that produces $\mathbf{K}(\lambda) = K$ and obtain f_λ in (2.67). Since $\mathbf{K}$ is decreasing, a simple linear search will produce the desired λ.

2.3.3.3 Optimal Plan: Bivariate Normal Prior, Exponential Detection Function, $B = \infty$

This example presents a case in which the optimal plan may be found explicitly.

Prior The search space S is the plane and $x = (x_1, x_2)$. The prior distribution on target location is given by

$$p_G(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left[-\frac{1}{2}\left(\frac{x_1^2}{\sigma_1^2} + \frac{x_2^2}{\sigma_2^2}\right)\right] \quad \text{for} \quad (x_1, x_2) \in S. \tag{2.73}$$

A plot of this function is shown in Fig. 2.2 for the case where $\sigma_1 = \sigma_2$. This is called a circular normal distribution.

Detection Function The search is conducted by a sensor that moves at speed v and has sweep width W. The detection function is exponential with

$$b(x, z) = 1 - \exp(-Wvz) \quad \text{for} \ \ x \in S \ \ \text{and} \ \ z \ge 0 \tag{2.74}$$

where z is search effort density measured in time searched per unit area. Note that Wvz has units of swept area per unit area.

Cost Cost is measured in search time, and $c(x) = 1$ for $x \in S$. $C(f)$ is the search time required to execute allocation f. If T is the amount of search time available, then T is the cost constraint.

Optimal Plan for Time *T* To find the optimal plan for search time T, we compute

$$\rho(x, z) = b'(x, z)\, p_G(x) = Wve^{-Wvz} p_G(x) = \frac{Wve^{-Wvz}}{2\pi\sigma_1\sigma_2} e^{-r^2(x)/2}$$

where

$$r^2(x) = \frac{x_1^2}{\sigma_1^2} + \frac{x_2^2}{\sigma_2^2} \ \ \text{and} \ \ x = (x_1, x_2). \tag{2.75}$$

For any real number a, define $a^+ = \max\{0, a\}$. Solving for $f_\lambda(x)$ such that $\lambda = \rho(x, f_\lambda(x)) = b'(x, f_\lambda(x))\, p_G(x)$, we obtain

$$f_\lambda(x) = \frac{1}{Wv}\left(\ln\left(\frac{Wv}{2\pi\sigma_1\sigma_2\lambda}\right) - r^2(x)/2\right)^+. \tag{2.76}$$

To compute $\mathbf{K}(\lambda)$, make the change of variables $x_1/\sigma_1 = \widehat{r}\cos\theta$ and $x_2/\sigma_2 = \widehat{r}\sin\theta$. Setting $r_0(\lambda) = [2\ln(Wv/2\pi\sigma_1\sigma_2\lambda)]^{1/2}$, we have

$$\begin{aligned}\mathbf{K}(\lambda) &= \tfrac{2\pi\sigma_1\sigma_2}{Wv}\int_0^{r_0(\lambda)}\left[\ln\left(\frac{Wv}{2\pi\sigma_1\sigma_2\lambda}\right)-\widehat{r}^2/2\right]\widehat{r}d\widehat{r} \\ &= \tfrac{\pi\sigma_1\sigma_2}{Wv}\left[\ln\left(\tfrac{2\pi\sigma_1\sigma_2\lambda}{Wv}\right)\right]^2.\end{aligned} \tag{2.77}$$

Using (2.77), we find λ to satisfy $T = \mathbf{K}(\lambda)$ and obtain the optimal plan $f_T^* = f_\lambda$ for T search time from (2.76). The result is

$$f_T^*(x) = \begin{cases} \frac{1}{Wv}\left(\left(\frac{WvT}{\pi\sigma_1\sigma_2}\right)^{\frac{1}{2}} - \frac{r^2(x)}{2}\right) & \text{for } r^2(x) \leq 2\left(\frac{WvT}{\pi\sigma_1\sigma_2}\right)^{\frac{1}{2}} \\ 0 & \text{for } r^2(x) > 2\left(\frac{WvT}{\pi\sigma_1\sigma_2}\right)^{\frac{1}{2}}. \end{cases} \tag{2.78}$$

As an example, consider a circular normal prior with mean at $(0,0)$ and $\sigma_1 = \sigma_2 = \sqrt{2}$. Assume $W = v = 1$. Figures 2.8 and 2.9 show the optimal search density f_T^* and the posterior given failure to detect for $T = 20$.

In Fig. 2.9, we see that the posterior density has been leveled in the region where search has been applied. In the region where no search has been applied, the posterior density is lower than in the searched region. This is characteristic of optimal plans for exponential detection functions that don't depend on x such as the one used for this example. If we think about search effort as being applied incrementally in time, the effort begins in the high probability region and expands out from that as the posterior in that region is reduced. This behavior does not hold

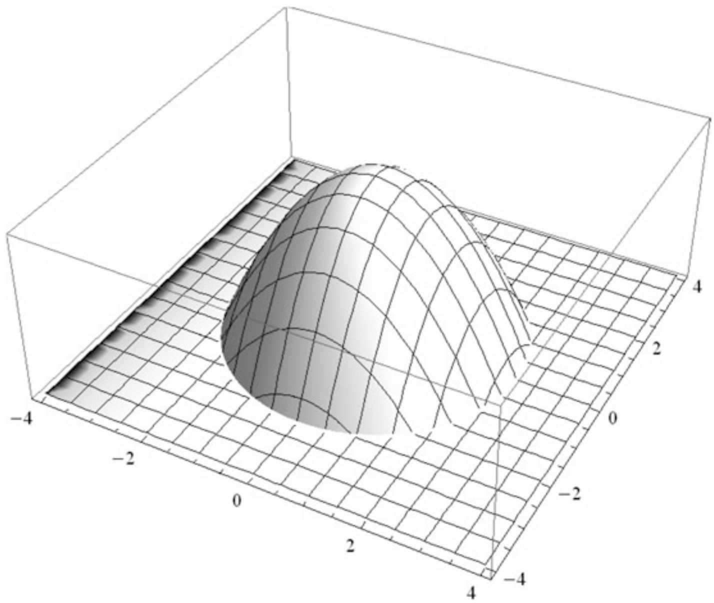


Fig. 2.8 Optimal search density at $T = 20$

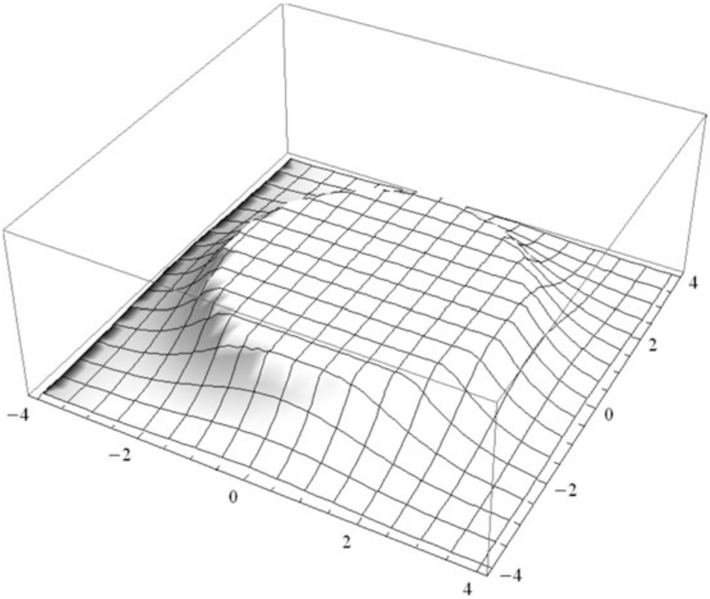


Fig. 2.9 Posterior probability density at time $T = 20$ given failure to detect

if the exponential detection function varies with x or if the detection function is not exponential as we shall see in Sect. 2.3.4.

Probability of Detection by Time T Examining (2.78), we see that the optimal plan f_T^* calls for an increasing amount of effort to be placed at each point x as T increases. Let us define a search plan φ in space and time so that $\varphi(x, T)$ gives the total search density that accumulates at x by time T in plan φ. Express the plan f_T^* in the polar coordinates used for the computation of $\mathbf{K}(\lambda)$ in (2.77), namely $x_1/\sigma_1 = \widehat{r}\cos\theta$ and $x_2/\sigma_2 = \widehat{r}\sin\theta$, so that

$$f_T^*\left(\widehat{r}\right) = \begin{cases} \frac{1}{Wv}\left(\left(\frac{WvT}{\pi\sigma_1\sigma_2}\right)^{\frac{1}{2}} - \frac{\widehat{r}^2}{2}\right) & \text{for } \widehat{r}^2 \le 2\left(\frac{WvT}{\pi\sigma_1\sigma_2}\right)^{\frac{1}{2}} \\ 0 & \text{for } \widehat{r}^2 > 2\left(\frac{WvT}{\pi\sigma_1\sigma_2}\right)^{\frac{1}{2}} \end{cases} \quad \text{for } 0 \le \theta \le 2\pi. \tag{2.79}$$

Under this transformation of variables, the prior density p_G becomes

$$\widehat{p}_G\left(\widehat{r}, \theta\right) = \frac{1}{2\pi}\exp^{-\widehat{r}^2/2} \text{ for } 0 \le \theta \le 2\pi,\ \widehat{r} \ge 0. \tag{2.80}$$

Set

$$\varphi^*\left(\left(\widehat{r}, \theta\right), T\right) = f_T^*\left(\widehat{r}\right) \text{ for } 0 \le \theta \le 2\pi,\ \widehat{r} \ge 0 \text{ and } T \ge 0. \tag{2.81}$$

Then the probability of detection by time T, $P\left(\varphi^*(\cdot, T)\right)$, is calculated by

$$
\begin{aligned}
P\left(\varphi^*(\cdot, T)\right) &= \int_0^{2\pi}\int_0^{\infty}\frac{1}{2\pi}e^{-\widehat{r}^2/2}\left[1-\exp\left(-Wv\varphi^*\left(\left(\widehat{r},\theta\right),T\right)\right)\right]\widehat{r}d\widehat{r}d\theta \\
&= 1-\int_0^{\infty}e^{-\widehat{r}^2/2}\exp\left(-Wvf_T^*\left(\widehat{r}\right)\right)\widehat{r}d\widehat{r}.
\end{aligned}
\tag{2.82}
$$

Let $H = \left(Wv/\pi\sigma_1\sigma_2\right)^{1/2}$ and $R^2(T) = 2H\sqrt{T}$, then

$$
Wvf_T^*\left(\widehat{r}\right) = \begin{cases} H\sqrt{T}-\widehat{r}^2/2 & \text{for } 0 \le \widehat{r} \le R(T) \\ 0 & \text{otherwise,}\end{cases}
$$

and

$$
\begin{aligned}
P\left(\varphi^*(\cdot, T)\right) &= 1-\int_0^{R(t)}e^{-H\sqrt{T}}\widehat{r}d\widehat{r}-\int_{R(t)}^{\infty}e^{-\widehat{r}^2/2}\widehat{r}d\widehat{r} \\
&= 1-e^{-H\sqrt{T}}\left(H\sqrt{T}+1\right) \quad \text{for } T \ge 0.
\end{aligned}
\tag{2.83}
$$

Mean Time to Detection Suppose we implement the time and space search plan φ^* given in (2.81) so that at each time T we obtain the detection probability given by (2.83). An integration by parts shows that the mean time to detect the target is

$$
\mu\left(\varphi^*\right) = \int_0^{\infty}t\frac{dP\left(\varphi^*(\cdot, t)\right)}{dt}dt = \int_0^{\infty}\left[1-P\left(\varphi^*(\cdot, t)\right)\right]dt = \frac{6\pi\sigma_1\sigma_2}{Wv}. \tag{2.84}
$$

Furthermore, since φ^* maximizes $P\left(\varphi^*(\cdot, t)\right)$ for all $t \ge 0$, it minimizes the mean time to detection.

The optimal allocation in (2.78) was originally found by Koopman (1946, 1957).

2.3.4 Optimal Plans with Uncertain Sweep Width

In the above example, we assumed that the sweep width W of the sensor is known. In some situations there is uncertainty about the sweep width. There may be uncertainty about the search sensor performance because it has not been tested against the target of interest, or we may not know the condition of the target, which can affect sensor performance. In this section we present an example of optimal search with uncertain sweep width in the case where the sweep width has a gamma distribution. This example is based on work in Richardson and Belkin (1972). Optimal search with uncertain sweep width is discussed in more detail and generality in Sect. 2.3 of Stone (2007).

Uncertain Sweep Width Model We model uncertainty in sweep width by specifying a probability distribution on sweep width W. This distribution can be discrete or continuous, but for the discussion here, we will assume that the uncertainty in sweep width is modeled by a probability density function. We assume that no information about the sweep width is obtained during the search except through failure to detect the target.

$$\Pr\{W = w\} = g(w) \text{ for } w \geq 0 \tag{2.85}$$

where we use Pr to mean probability density.

Detection Function For each value of w, there is a decreasing-rate detection function $b_w(x, z)$. This is the conditional detection function. The one that holds conditioned on $W = w$. The unconditional detection function is given by

$$b\,(x, z) = \int_0^\infty b_w\,(x, z)\, g(w) dw. \tag{2.86}$$

Optimal Plan Section 2.3 of Stone (2007) shows that if $b_w(x, z)$ is a decreasing-rate detection function for all w, then the unconditional detection function $b(x, z)$ defined in (2.86) is also a decreasing-rate detection function. (In Stone (2007) this is called a regular detection function.) As a result, we may find optimal plans for uncertain sweep width by calculating the unconditional detection function and computing the optimal plans from (2.31) and (2.32) for discrete spaces or from (2.71) and (2.72) for continuous spaces. We now provide an example of computing the optimal plan for uncertain sweep width.

Example: Optimal Plan for a Gamma-Distributed Uncertain Sweep Width For this example we assume that the sweep width has an gamma distribution, i.e.,

$$\Pr\{W = w\} = g(w) = \frac{w^{\alpha-1} \beta^{\alpha} e^{-\beta w}}{\Gamma\,(\alpha)} \text{ for } w \geq 0 \tag{2.87}$$

where $\alpha > 0$, $\beta > 0$ and Γ is the standard gamma function for which $\Gamma\,(\alpha) = (\alpha - 1)!$ for integer values of α. The mean of a gamma distribution is α/β and the variance is α/β^2. The ratio of the standard deviation to the mean is $1/\sqrt{\alpha}$ which approaches 0 as $\alpha \to \infty$.

Examples of gamma densities are shown in Fig. 2.10 were we have held the mean fixed at 1 and let α increase. The resulting densities become more peaked about the mean as α increases. This reflects a situation with a fixed mean for the sweep width, and a range of possible uncertainties.

We assume the conditional detection function is

$$b_w\,(x, z) = 1 - e^{-wvz} \text{ given } W = w. \tag{2.88}$$

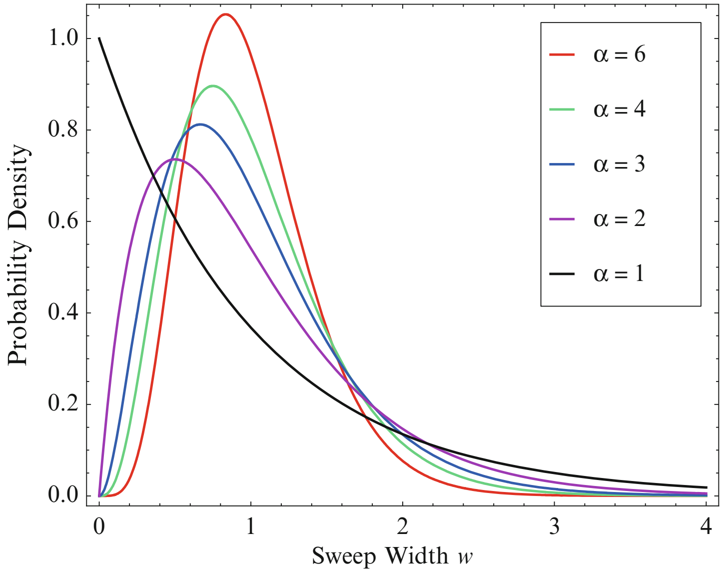


Fig. 2.10 Gamma densities with mean $\alpha/\beta = 1$

The conditional detection function is the same as the exponential detection function in (2.74) with $W = w$. From (2.87), we calculate

$$b(x,z) = \int_0^{\infty} (1 - e^{-wvz})\, g(w)dw = 1 - \left(1 + \frac{vz}{\beta}\right)^{-\alpha} \quad \text{for } z \geq 0$$

and

$$b'(x,z) = v\alpha\beta^{\alpha}(\beta + vz)^{-(\alpha+1)} \text{ for } z \geq 0. \tag{2.89}$$

From (2.89) we obtain

$$\rho(x,z) = p_G(x) v\alpha\beta^{\alpha}(\beta + vz)^{-(\alpha+1)} \text{ for } z \geq 0,$$

and for $\lambda > 0$

$$\rho^{-1}(x,\lambda) = \begin{cases} \frac{\beta}{v}\left[\left(p_G(x)v\alpha/\lambda\beta\right)^{1/(\alpha+1)} - 1\right] & \text{for } p_G(x) \geq \lambda\beta/(v\alpha) \\ 0 & \text{for } p_G(x) < \lambda\beta/(v\alpha). \end{cases} \tag{2.90}$$

Let

$$D = \frac{2\pi\sigma_1\sigma_2\beta}{v\alpha}.$$

Then we may rewrite (2.90) as

$$\rho^{-1}(x,\lambda) = \begin{cases} \frac{\beta}{v}\left[\left(p_G(x)v\alpha/\lambda\beta\right)^{1/(\alpha+1)} - 1\right] & \text{for } r^2(x) \le -2\ln(D\lambda) \\ 0 & \text{for } r^2(x) > -2\ln(D\lambda). \end{cases} \tag{2.91}$$

Making the change of variables $x_1/\sigma_1 = \widehat{r}\cos\theta$ and $x_2/\sigma_2 = \widehat{r}\sin\theta$ that we used in (2.77), we obtain $\widehat{p}_G$ in (2.80) and from (2.90), we have

$$\begin{aligned} \mathbf{K}(\lambda) &= 2\pi\sigma_1\sigma_2\int_0^{\sqrt{-2\ln(D\lambda)}} \frac{\beta}{v}\left[\left(\frac{\widehat{p}_G(\widehat{r})\,v\alpha}{\sigma_1\sigma_2\beta\lambda}\right)^{1/(\alpha+1)} - 1\right]\widehat{r}d\widehat{r} \\ &= \alpha D\int_0^{\sqrt{-2\ln(D\lambda)}}\left[\left(\frac{e^{-\widehat{r}^{2/2}}}{D\lambda}\right)^{1/(\alpha+1)} - 1\right]\widehat{r}d\widehat{r} \\ &= \alpha(\alpha+1)D\left[(D\lambda)^{-1/(\alpha+1)} + \tfrac{\ln(D\lambda)}{\alpha+1} - 1\right] \quad \text{for } 0 < \lambda \le 1/D. \end{aligned} \tag{2.92}$$

From (2.72) and (2.91), we can express the optimal time and space search plan $\widehat{\varphi}^*$ for uncertain sweep width in terms of $\lambda(t) = \mathbf{K}^{-1}(t)$ as follows.

$$\widehat{\varphi}^*(x,t) = \begin{cases} \frac{\beta}{v}\left[\left(e^{-r^2(x)/2}/D\lambda(t)\right)^{1/(\alpha+1)} - 1\right] & \text{for } r^2(x) \le -2\ln(D\lambda(t)) \\ 0 & \text{for } r^2(x) > -2\ln(D\lambda(t)) \end{cases} \tag{2.93}$$

where $r^2(x) = x_1^2/\sigma_1^2 + x_2^2/\sigma_2^2$ as before.

We now compute the optimal uncertain-sweep-width plan for time $T = 20$ in the case where $\alpha = \beta = 2, \sigma_1 = \sigma_2 = \sqrt{2}$, and $v = 1$. This example is the same as the one in Sect. 2.3.3.3, except that the certain sweep $W = 1$ of that example is replaced with an uncertain sweep width having a gamma distribution with parameters $\alpha = 2$, $\beta = 2$, and mean 1. This distribution has the probability density function given by the $\alpha = 2$ curve in Fig. 2.10.

Since we do not have an explicit form for $\mathbf{K}^{-1}$, we solve (2.92) numerically to obtain the value $\lambda(20)$ for which $\mathbf{K}(\lambda(20)) = 20$. We use this value in (2.93) to compute the optimal uncertain-sweep-width search density which is shown in Fig. 2.11 It looks similar to the optimal certain-sweep-width search density in Fig. 2.8 for $T = 20$ and $W = 1$. However, the radius at which the density for the certain-sweep-width plan reaches 0 is 2.67; whereas the density for the uncertain-sweep-width plan reaches 0 at radius 2.8. One effect of the uncertainty in sweep width is to increase the spread of the search area, at least initially. See Belkin (1975).

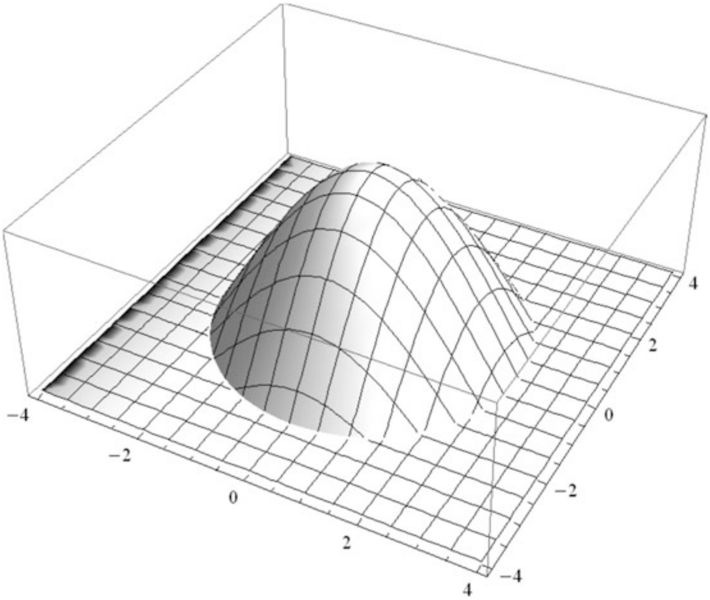


Fig. 2.11 Optimal search density for uncertain sweep width:$T = 20, \alpha = \beta = 2, \sigma_1 = \sigma_2 = \sqrt{2}, v = 1$

Figure 2.12 shows the posterior target probability density given failure to detect by $T = 20$. This posterior is not level in the region where search has taken place as is the case for the certain sweep width example.

Mean Time to Detection for Optimal Uncertain Sweep Width Plan Following the method given in Sect. 2.3 of Stone (2007), we may calculate the mean time to detection for the optimal uncertain sweep width plan and find

$$\mu\left(\widehat{\varphi}^*\right) = \begin{cases} \frac{2\pi\sigma_1\sigma_2\beta}{v\alpha}\left(\frac{3\alpha+1}{\alpha-1}\right) & \text{for } \alpha > 1 \\ \infty & \text{for } 0 < \alpha \leq 1. \end{cases} \tag{2.94}$$

To see the effect of uncertain sweep width on mean time to detection, let us set $W = \alpha/\beta$ in (2.84) to obtain the mean time to detect for a plan with known sweep equal to the mean of the gamma sweep width distribution. Doing this we obtain

$$\mu\left(\varphi^*\right) = \frac{6\pi\sigma_1\sigma_2\beta}{\alpha v} \text{ for } \alpha > 1.$$

From this we obtain

$$\frac{\mu\left(\widehat{\varphi}^*\right)}{\mu\left(\varphi^*\right)} = \frac{1}{3}\left(\frac{3\alpha+1}{\alpha-1}\right) \geq 1 \text{ for } \alpha > 1 \tag{2.95}$$

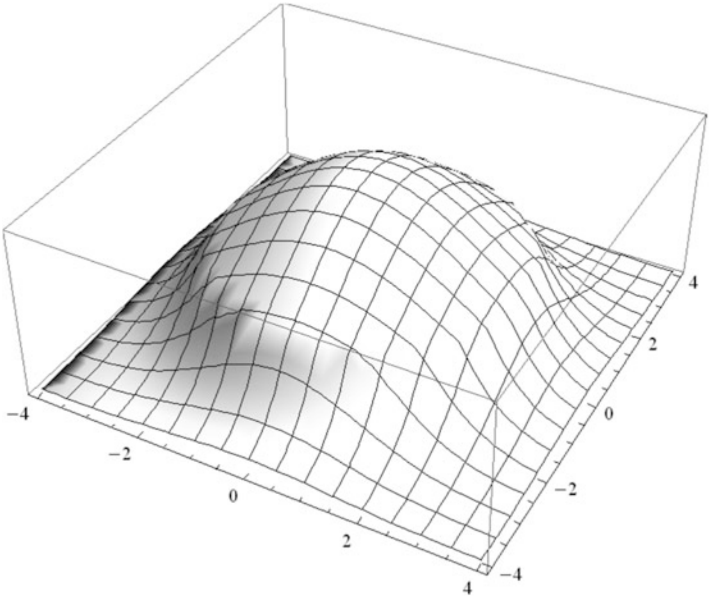


Fig. 2.12 Posterior target probability density given failure to detect: $T = 20, \alpha = \beta = 2, \sigma_1 = \sigma_2 = \sqrt{2}, v = 1$

and see that one pays a penalty in mean time to detection for uncertainty in the sweep width. Notice that as $\alpha \to \infty$ in (2.95), $\mu(\widehat{\varphi}^*)/\mu(\varphi^*) \to 1$. This reflects the fact that as α increases, the spread of the gamma distribution about its mean decreases more and more.

2.3.5 Uniformly Optimal Search Plans

So far we have discussed plans that are optimal for a fixed cost or search time. Often we do not know the exact length of time available for search. In this case, it would be desirable to follow a search allocation in space and time that produces the maximum detection probability at each time $t \geq 0$.

Suppose that $M(t) \geq 0$ total search effort (cost) is available for $t \geq 0$, where M is an increasing function of t. We define the class $\Phi(M)$ of continuous space and time allocation functions as follows:

$$\begin{aligned}
&\varphi \in \Phi(M) \text{ if and only if} \\
&\varphi(x,t) \geq 0 \text{ for } x \in S,\ t \geq 0 \\
&\varphi(x,t) \text{ is an increasing function of } t \text{ for } x \in S \\
&\int_S \varphi(x,t)\,dx = M(t) \text{ for } t \geq 0.
\end{aligned} \tag{2.96}$$

For discrete space, we have an analogous definition.

$$\begin{aligned}&\varphi \in \Phi(M) \text{ if and only if}\\&\varphi\,(j,t) \geq 0 \text{ for } j=1,\ldots,J,\ \ t \geq 0\\&\varphi\,(j,t)\ \text{ is an increasing function of } t \text{ for } 1 \leq j \leq J\\&\sum\nolimits_{j=1}^{J} \varphi\,(j,t) = M(t) \text{ for } t \geq 0.\end{aligned} \tag{2.97}$$

A plan $\varphi^* \in \Phi(M)$ is *uniformly optimal in $\Phi(M)$* if and only if

$$P\left(\varphi^*\,(\cdot,t)\right)\ \text{ is optimal for cost M}(t) \text{ for all } t \geq 0. \tag{2.98}$$

The algorithms for finding optimal search plans in Sects. 2.3.1.2 and 2.3.3.2 provide a method of finding uniformly optimal plans for discrete and continuous search spaces when search effort is continuous and the detection function has a decreasing rate.

Uniformly Optimal Plan: Discrete Space, Continuous Effort Assume that the detection function is a decreasing-rate detection function. Following (2.31) and (2.32), we let

$$\begin{aligned}&\lambda\,(M(t)) = \mathbf{K}^{-1}\,(M(t))\ \text{ for } t \geq 0\\&\varphi^*\,(j,t) = \rho^{-1}\,(j,\lambda\,(M(t)))\ \text{ for } 1 \leq j \leq J \text{ and } t \geq 0.\end{aligned} \tag{2.99}$$

Since M is increasing, $\lambda(M(t))$ is a decreasing function of t. As a result, $\varphi^*\,(j,\cdot)$ is an increasing function of t for $1 \leq j \leq J$. In addition,

$$\sum\nolimits_{j=1}^{J} \varphi\,(j,t) = M(t) \text{ for } t \geq 0,$$

so that $\varphi^* \in \Phi(m)$. By the construction of φ^*, using the algorithm in (2.31) and (2.32),

$$P\left(\varphi^*\,(\cdot,t)\right)\ \text{ is optimal for cost } M(t) \text{ for all } t \geq 0,$$

and φ^* is uniformly optimal in $\Phi(M)$.

Uniformly Optimal Plan: Continuous Space, Continuous Effort Assume that the detection function is decreasing-rate. Following (2.71) and (2.72), we let

$$\begin{aligned}&\lambda\,(M(t)) = \mathbf{K}^{-1}\,(M(t))\ \text{ for } t \geq 0\\&\varphi^*\,(x,t) = \rho^{-1}\,(x,\lambda\,(M(t)))\ \text{ for } x \in X \text{ and } t \geq 0\end{aligned} \tag{2.100}$$

to obtain the plan φ^* which is uniformly optimal in $\Phi(m)$.

As we have noted above, a uniformly optimal plan minimizes the mean time (cost) to detection. Theorem 2.4.6 of Stone (2007) shows that a uniformly optimal plan exists for a continuous space, continuous effort search whenever the detection function $b(x, z)$ is an increasing, continuous function of z and $c(x)$ equals a constant $\overline{c} > 0$ for $x \in X$. The same result does not hold for a discrete space search. To guarantee the existence of a uniformly optimal plan for discrete space, one must add the assumption that $b\,(j, \cdot)$ is concave for $1 \leq j \leq J$.

Uniformly Optimal Plan: Discrete Space and Effort The plan φ^* constructed in Sect. 2.3.2.1 is uniformly optimal within the class of plans that allocate one look at each time period. Thus φ^* minimizes the expect cost to detect the target.

2.4 Defective Prior Distributions

In some cases, we may not be sure that the target is in the search region. In this case, it is appropriate to have a prior that sums or integrates to a probability, $p_0 < 1$. The optimal search plan for a defective prior is the same as the one we would obtain by scaling the defective prior so that it sums or integrates to 1. The difference is that the success probability is scaled down by the factor p_0. Thus one can use the algorithms given in this chapter to find optimal plans for defective priors by simply scaling the prior to sum or integrate 1 and applying the algorithm to the rescaled prior. As mentioned above, the resulting probability of success needs to be multiplied by p_0 to account for the fact the target may not be in the search region.

2.5 Summary

This chapter has presented the basic results on optimal search for a stationary target. Section 2.1 presented examples of prior distributions on discrete and continuous state spaces. Section 2.2 defined detection functions, lateral range functions, and sweep width. It derived the exponential and inverse cube detection models. Section 2.3 defined the basic problem of optimal search for a stationary target and developed algorithms for computing optimal plans for searches involving decreasing-rate detection functions in the case of continuous effort for both discrete and continuous state spaces and in the case of discrete effort for discrete state spaces. Section 2.3.4 showed that these algorithms can be extended to the case where the sweep width of the sensor is uncertain. Examples of optimal plans were found for a bivariate normal distribution in the case of an exponential detection function with a known sweep width and for the case with an uncertain sweep width. Section 2.3.5 defined uniformly optimal plans and showed that the algorithms for computing optimal plans for decreasing-rate detection functions can be used to compute uniformly optimal plans. Uniformly optimal plans specify search allocations in space and time. They maximize detection probability at each time and therefore minimize mean time to detection.

2.6 Notes

Stone (2007) covers a much wider range of stationary target problems than are presented here and goes into more detail about their properties. As an example, it shows that for concave detection functions, incrementally optimal search plans are totally optimal. This means that if a search is planned in increments in such a fashion that each increment of effort is allocated to maximize the increase in detection probability given the failure of the previous increments, then the plan that results from these increments will be optimal for the total effort in the increments. In the case of a continuous state space, incrementally optimal is totally optimal for any detection function.

In the case of discrete effort, Stone (2007) finds optimal whereabouts plans. A whereabouts plans specifies a sequence of cells in which to search. If this search does not detect the target, the plan is allowed to guess one cell for the location of the target. If the plan either detects the target or guesses right, it succeeds. Chapter 5 considers optimal search and stop problems where there is a cost to searching and a reward if the target is found. The goal is to find a plan that tells the searcher where to search and when to stop in order to maximize expected return. There is a chapter devoted to approximating optimal plans with simpler ones that are more operationally feasible than optimal ones. There is also a chapter on optimal search in the presence of false targets. The notes that accompany the chapters give credit for the results in search for stationary targets and provide some history. Washburn (2006) investigates piled-slab plans that are simpler to implement than the optimal ones but that still provide good approximations to the detection probability of the optimal plan.

References

Belkin B (1975) On the rate of expansion of gamma search plans. Oper Res 23:1033–1037

Charnes A, Cooper WW (1958) The theory of search: optimum distribution of search effort. Manag Sci 5:44–50

Coast Guard (2010) U.S. Coast Guard addendum to the United States search and rescue supplement (NSS) to the international aeronautical and maritime search and rescue manual (IAMSAR). COMDTINST M16130.2F, U.S., Coast Guard, Washington, DC

Everett H (1963) Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. Oper Res 11:399–417

IAMSAR (2008) International aeronautical and maritime search and rescue manual, vol 2. IMO/ICAO, London/Montréal

Koopman BO (1946) Search and screening, vol 56, Operations evaluation group report. Center for Naval Analyses, Alexandria

Koopman BO (1956a) The theory of search. Part I. Kinematic bases. Oper Res 4:324–346

Koopman BO (1956b) The theory of search. Part II. Target detection. Oper Res 4:503–531

Koopman BO (1957) The theory of search. Part III. The optimum distribution of searching effort. Oper Res 5:613–626

Koopman BO (1980) Search and screening: general principles with historical applications. Pergamon Press, New York

Kratzke TM, Stone LD, Frost JR (2010) Search and rescue optimal planning system. In: Proceedings of the 13th international conference on information fusion, Edinburgh, UK, 26–29 July 2010

Richardson HR, Belkin B (1972) Optimal search with uncertain sweep width. Oper Res 20:764–784

Stone LD (1992) Search for the *SS Central America*: mathematical treasure hunting. Interfaces 22:32–54

Stone LD (2007) Theory of optimal search, 2nd edn. INFORMS, Catonsville

Washburn AR (2006) Piled-slab searches. Oper Res 54:1193–1200

Washburn AR (2014a) Search and detection, 5th edn. Create Space, North Charleston

Chapter 3
Search for a Moving Target in Discrete Space and Time

This chapter develops methods for finding optimal search plans for a target that is moving in discrete space and time. In the case where the detection function is exponential, optimal moving target plans can be obtained by computing a sequence of optimal stationary target plans. The algorithm developed to find these plans is a special case of the more general Forward-And-Backward (FAB) algorithm which is also presented in this chapter.

3.1 Continuous Effort Search Problem

For convenience, we consider the search to take place over an interval of time $[0, T]$. We represent times by integers so that search times t are numbered $t = 0, \ldots, T$. The increments between times t and $t + 1$ for $t = 0, \ldots, T - 1$ need not be equal. The discrete state space is the set of J cells as in Chap. 2.

Prior Distribution The prior distribution for a moving target problem in discrete space and time specifies the target's motion through space and time as a stochastic process $X = \{X(t),\ t = 0, \ldots, T\}$ where $X(t)$ is the target's state (cell) at time t. Let $\omega = (\omega_0, \ldots, \omega_T)$ be a sample path of the process, i.e., ω_t is the cell the target occupies at time t for $t = 0, \ldots, T$. Let Ω be the set of sample paths of X and let p be the probability distribution on the paths. Specifically,

$$p(\omega) = \Pr\{\omega = (\omega_0, \ldots, \omega_T) \text{ is the target's path}\} \text{ for } \omega \in \Omega \tag{3.1}$$

Electronic supplementary material The online version of this chapter (doi: 10.1007/978-3-319-26899-6_3) contains supplementary material, which is available to authorized users.

L.D. Stone et al., *Optimal Search for Moving Targets*, International Series in Operations Research & Management Science 237,
DOI 10.1007/978-3-319-26899-6_3

where

$$\sum_{\omega \in \Omega} p(\omega) = 1.$$

Continuous Effort Search Plan A continuous effort search plan is a space-time search allocation f where $f(j, t)$ is the effort placed in cell j at time t. We suppose that $m(t) \geq 0$ search effort is available for $t \geq 0$. We define the class $F(m)$ of continuous-effort, discrete-space-and-time search plans as follows.

$$\begin{aligned} &f \in F(m) \text{ if and only if} \\ &0 \leq f(j, t) < \infty \text{ for } 1 \leq j \leq J \text{ and } t = 0, \ldots, T \\ &\sum\nolimits_{j=1}^{J} f(j, t) = m(t) \text{ for } t = 0, \ldots, T. \end{aligned} \tag{3.2}$$

There may be an upper bound B on the search density. In this case we define the class $F_B(m)$ of search plans where

$$\begin{aligned} &f \in F_B(m) \text{ if and only if} \\ &\qquad f \in F(m) \\ &\qquad f(x, t) \leq B \text{ for } x \in S \text{ and } t = 0, \ldots, T. \end{aligned} \tag{3.3}$$

If $B = \infty$, then $F_B(m) = F(m)$.

Detection Function The probability of detecting the target, given it follows path ω, is a function of the weighted total search effort that "falls on" the target as it follows the path ω. The function

$$\zeta(f, \omega, t) = \sum\nolimits_{s=0}^{t} W(\omega_s, s) f(\omega_s, s) \text{ for } \omega \in \Omega,\ t = 0, \ldots, T \tag{3.4}$$

accumulates the weighted search effort over $[0, t]$ for the path ω where the weight $W(j, s)$ represents the relative detectability or sweep width for the target given it is located in cell j at time s. There is a detection function b such that

$$b(\zeta(f, \omega, t)) = \Pr\{\text{detecting the target by time } t | \text{target follows path } \omega\} \tag{3.5}$$

Probability of Detection The probability of detection by time t for a plan $f \in F_B(m)$ is

$$P(f, t) = E[b(\zeta(f, \omega, t))] \text{ for } t = 0, \ldots, T \tag{3.6}$$

where $E[\,]$ indicates expectation over the probability distribution p on the sample paths of X. We can write (3.6) as

$$P(f, t) = \sum_{\omega \in \Omega} p(\omega)\, b\left(\sum\nolimits_{s=0}^{t} W(\omega_s, s) f(\omega_s, s)\right) \text{ for } f \in F(m). \tag{3.7}$$

***T*-Optimal Plan** A plan $f^* \in F_B(m)$ is T-*optimal* if and only if

$$P\left(f^*, T\right) \geq P\left(f, T\right) \text{ for } f \in F_B(m). \tag{3.8}$$

In most cases, we cannot construct a moving target plan that is uniformly optimal. This means that the plan that is optimal for $T + \Delta \mathrm{T}$ is not an extension of the T-optimal plan. One has to choose the time at which he wishes the plan to be optimal. This makes it more difficult to find plans that minimize mean time to detection.

3.1.1 Necessary and Sufficient Conditions for a T-Optimal Plan: Continuous-Effort, Decreasing-Rate Detection Function

In this section we find necessary and sufficient conditions for a plan $f^* \in F_B(m)$ to be T-Optimal when b is a decreasing-rate detection function.

Restatement of Optimization Problem Let us consider a space-time allocation plan f as a $J(T+1)$ vector where each component takes values in $[0, \infty)$. To obtain the necessary and sufficient conditions for a plan to be T-Optimal, it is convenient to restate the T-optimal search problem in a more standard form. Namely,

Find f to

$$\text{maximize } P\left(f, T\right) \tag{3.9}$$

Subject to the following constriants

$$\begin{aligned} &0 \leq f\left(j, t\right) \leq B \text{ for } j = 1, \ldots, J \text{ and } t = 0, \ldots, T \\ &\sum_{j=1}^{J} f\left(j, t\right) = m(t) \text{ for } t = 0, \ldots, T. \end{aligned} \tag{3.10}$$

Since b is a decreasing-rate detection function, it is concave on $[0, \infty)$. As a result, each term in (3.7) is concave, and $P(f, T)$ is a concave function of f for $f \in F_b(m)$. Since the constraints in (3.10) are linear and $P\left(\cdot, T\right)$ is concave and differentiable, the Karush-Kuhn-Tucker conditions (Bertsekas 1999 Chap. 3) are satisfied. These conditions yield the following result.

Theorem 3.1: Discrete Space and Time Optimality Conditions *Assume b is a decreasing-rate detection function.* Then $f^* \in F_B(m)$ *solves the optimization problem in (3.9)–(3.10) if and only if there exists a vector* $(\lambda_0, \ldots, \lambda_T)$ *with positive components such that for* $t = 0, \ldots, T$

$$\begin{aligned} \frac{\partial P\left(f^*, T\right)}{\partial f\left(j, t\right)} &\geq \lambda_t \text{ if } f^*\left(j, t\right) = B \\ &= \lambda_t \text{ if } 0 < f^*\left(j, t\right) < B \text{ for } j = 1, \ldots, J. \\ &\leq \lambda_t \text{ if } f^*\left(j, t\right) = 0 \end{aligned} \tag{3.11}$$

3.1.2 Bound on Optimal Plan: Continuous-Effort, Decreasing-Rate Detection Function

In this section we find an upper bound on the probability of detection for the T-optimal plan. This bound will be useful in providing a stopping criterion for algorithms that find approximations to optimal plans. The bound relies on the observation that if b is a decreasing-rate detection function then

$$b(z_2) - b(z_1) \le b'(z_1)(z_2 - z_1) \text{ for } z_1, z_2 \ge 0. \tag{3.12}$$

Let $f_1, f_2 \in F_B(m)$ be two allocation functions. Then from (3.12)

$$\begin{aligned} P(f_2, T) - P(f_1, T) &= E[b(\zeta(f_2, \omega, T))] - E[b(\zeta(f_1, \omega, T))] \\ &\le E\left[b'(\zeta(f_1, \omega, T))(\zeta(f_2, \omega, T) - \zeta(f_1, \omega, T))\right]. \end{aligned} \tag{3.13}$$

Expanding the right-hand side of (3.13), we obtain

$$\begin{aligned} &E[b'(\zeta(f_1, \omega, T))(\zeta(f_2, \omega, T) - \zeta(f_1, \omega, T))] \\ &\quad = \sum_{t=0}^{T} E\left[b'(\zeta(f_1, \omega, T))\, W(\omega_t, t)(f_2(\omega_t, t) - f_1(\omega_t, t))\right]. \end{aligned} \tag{3.14}$$

Let E_{jt} indicate expectation conditioned on $X(t) = j$ and $p_t(j) = \Pr\{X(t) = j\}$ for $j = 1, \ldots, J,\ t = 0, \ldots, T$. Define

$$D(f, j, t) = E_{jt}\left[b'(\zeta(f, \omega, T))\right] p_t(j) W(j, t) \text{ for } f \in F(m). \tag{3.15}$$

Observe that $D(f, j, t) \ge 0$. Since

$$\begin{aligned} &E[b'(\zeta(f_1, \omega, T))\, W(\omega_t, t)(f_2(\omega_t, t) - f_1(\omega_t, t))] \\ &\quad = \sum_{j=1}^{J} E_{jt}[b'(\zeta(f_1, \omega, T))]\, p_t(j) W(j, t)(f_2(j, t) - f_1(j, t)), \end{aligned}$$

we can write (3.14) as

$$\begin{aligned} &E[b'(\zeta(f_1, \omega, T))(\zeta(f_2, \omega, T) - \zeta(f_1, \omega, T))] \\ &\quad = \sum_{t=0}^{T} \sum_{j=1}^{J} D(f_1, j, t)(f_2(j, t) - f_1(j, t)). \end{aligned} \tag{3.16}$$

Let

$$\left.\begin{aligned} \underline{\lambda}(t) &= \begin{cases} \min\limits_{j=1,\ldots,J} D(f_1, j, t) & \text{if } f_1(j, t) > 0 \text{ for some } j \\ 0 & \text{otherwise} \end{cases} \\ \overline{\lambda}(t) &= \begin{cases} \max\limits_{j=1,\ldots,J} D(f_1, j, t) & \text{if } f_1(j, t) < B \text{ for some } j \\ \underline{\lambda}(t) & \text{otherwise} \end{cases} \end{aligned}\right\} \text{ for } t = 0, \ldots, T. \tag{3.17}$$

Let

$$\mathcal{T} = \{t : f_1(j,t) = B \text{ for } j = 1, \ldots, J\}.$$

Note that for $t \in \mathcal{T}$, $\overline{\lambda}(t) = \underline{\lambda}(t)$, and since $f_1, f_2 \in F_B(m)$, $m(t) = BJ$ and $f_1(j,t) = f_2(j,t) = B$ for $j = 1, \ldots, J$. From (3.16) and (3.17), we have

$$\begin{aligned}
&E\left[b'\left(\zeta\left(f_1,\omega,T\right)\right)\left(\zeta\left(f_2,\omega,T\right) - \zeta\left(f_1,\omega,T\right)\right)\right] \\
&= \sum_{t \notin \mathcal{T}} \sum\nolimits_{j=1}^{J} D\left(f_1, j, t\right)\left(f_2(j,t) - f_1(j,t)\right) \\
&\leq \sum_{t \notin \mathcal{T}} \sum\nolimits_{j=1}^{J} \left[\overline{\lambda}(t) f_2(j,t) - \underline{\lambda}(t) f_1(j,t)\right] \\
&= \sum_{t \notin \mathcal{T}} \left(\overline{\lambda}(t) - \underline{\lambda}(t)\right) m(t) \\
&= \sum\nolimits_{t=0}^{T} \left(\overline{\lambda}(t) - \underline{\lambda}(t)\right) m(t).
\end{aligned} \tag{3.18}$$

From (3.13) and (3.18), we have

$$P\left(f_2, T\right) - P\left(f_1, T\right) \leq \sum\nolimits_{t=0}^{T} \left(\overline{\lambda}(t) - \underline{\lambda}(t)\right) m(t) \equiv \Delta\left(f_1\right) \text{ for } f_1, f_2 \in F_B(m). \tag{3.19}$$

Notice the right-hand side of (3.19) does not depend on f_2. Thus if $f^* \in F_B(m)$ is T-optimal then

$$P\left(f^*, T\right) \leq P(f, T) + \Delta(f) \text{ for any } f \in F_B(m).$$

We can now state the upper bound theorem obtained by Washburn (1981).

Theorem 3.2 *If b is an decreasing-rate detection function and $f^* \in F_B(m)$ is T-optimal, then*

$$P\left(f^*, T\right) \leq P(f, T) + \Delta(f) \text{ for any } f \in F_B(m) \tag{3.20}$$

where

$$\Delta\,(\mathrm{f}) = \sum\nolimits_{t=0}^{T} \left(\overline{\lambda}(t) - \underline{\lambda}(t)\right) m(t).$$

3.2 Optimal Plans: Continuous-Effort, Exponential Detection Function

When the detection function is exponential, we can show that finding a T-optimal search plan is equivalent to solving a sequence of stationary target problems. This means we can use the algorithms developed in Chap. 2 for stationary targets to find optimal plans for moving targets.

Suppose the detection function $b(z) = 1 - e^{-z}$ for $z \geq 0$ and f^* is T-optimal in $F_B(m)$. Then for any t we can rewrite (3.7) as

$$\begin{aligned} 1 - P(f^*, T) &= \sum_{j=1}^{J} \sum_{\{\omega:\omega_t=j\}} p(\omega) \exp\left(-\sum\nolimits_{s=0}^{T} W(\omega_s, s) f^*(\omega_s, s)\right) \\ &= \sum_{j=1}^{J} e^{-W(j,t)f^*(j,t)} \sum_{\{\omega:\omega_t=j\}} p(\omega) \exp\left(-\sum_{s\neq t} W(\omega_s, s) f^*(\omega_s, s)\right). \end{aligned} \tag{3.21}$$

Let

$$q(j,t,f) = \sum_{\{\omega:\omega_t=j\}} p(\omega) \exp\left(-\sum_{s\neq t} W(\omega_s, s) f(\omega_s, s)\right). \tag{3.22}$$

Then (3.21) becomes

$$1 - P\left(f^*, T\right) = \sum_{j=1}^{J} e^{-W(j,t)f^*(j,t)} q\left(j,t,f^*\right) \text{ for } t = 0, \ldots, T. \tag{3.23}$$

Note that $q(j,t,f^*)$ equals the probability that the target is in cell j at time t and is not detected by the search at any time other than t. Thus $q(\cdot,t,f^*)$ is proportional to the posterior probability distribution $\tilde{q}(\cdot,t,f^*)$ on the target's location at time t given failure to detect at all times other than t. If f^* is T-optimal, then obviously it minimizes the failure probability $1 - P(f^*, T)$. From (3.23) it is clear that $f^*(\cdot,t)$ must minimize the failure probability for the stationary target search with distribution $\tilde{q}(\cdot,t,f^*)$ and effort $m(t)$ for $t = 0, \ldots, T$.

Now suppose that f^* is a moving target plan in $F_B(m)$ such that $f^*(\cdot,t)$ is an optimal stationary target plan for $\tilde{q}(\cdot,t,f^*)$ for $m(t)$ effort for $t = 0, \ldots, T$. From the discussion of defective distributions in Sect. 2.4, we know that f^* is also optimal for $q(\cdot,t,f^*)$. Define

$$Q(t,g) = 1 - \sum\nolimits_{j=1}^{J} e^{-W(j,t)g(j)} q\left(j,t,f^*\right) \text{ for } t = 0, \ldots, T. \tag{3.24}$$

Then $Q(t,g)$ is the probability of detection produced by the stationary target plan g applied to the distribution $q(\cdot,t,f^*)$. Since $f^*(\cdot,t)$ maximizes this probability, we have from Theorem 3.1 (applied to a single time period search) that there is a $\lambda_t > 0$ such that

$$\begin{aligned} &\frac{\partial Q(t, f^*(\cdot,t))}{\partial g(j)} \\ &\quad = W(j,t)\, e^{-W(j,t)f^*(j,t)} q(j,t,f^*) \geq \lambda_t \text{ if } f^*(j,t) = B \\ &\quad\qquad\qquad\qquad\qquad\qquad\qquad\quad = \lambda_t \text{ if } 0 < f^*(j,t) < B \\ &\quad\qquad\qquad\qquad\qquad\qquad\qquad\quad \leq \lambda_t \text{ if } f^*(j,t) = 0. \end{aligned} \tag{3.25}$$

From (3.7), we have

$$\begin{aligned}\frac{\partial P(f^*,T)}{\partial f(j,t)} &= W(j,t) \sum_{\{\omega:\omega_t=j\}} p(\omega) \exp\left(-\sum_{s=0}^{T} W(\omega_s,s) f^*(\omega_s,s)\right) \\ &= W(j,t)\, e^{-W(j,t)f^*(j,t)} q(j,t,f^*)\,. \end{aligned} \tag{3.26}$$

The summation in (3.26) includes only those paths that are in cell j at time t because P does not depend on $f(j,t)$ for the other paths. Equations (3.25) and (3.26) imply that conditions (3.11) hold for $t = 0,\ldots,T$ which in turn implies that f^* is T-optimal within $F_B(m)$. We now state these results as a theorem.

Theorem 3.3 *Assume b is an exponential detection function. Then a necessary and sufficient condition for $f^* \in F_B(m)$ to be a T-optimal plan is that $f^*(\cdot,t)$ is the optimal stationary target plan for cost $m(t)$ for the distribution $\tilde{q}(\cdot,t,f^*)$, which is the posterior probability distribution on the target's location at time t given failure to detect at all times other than t, for $t = 0,\ldots,T$.*

3.2.1 T-Optimal Search Plan Recursion for a Continuous-Effort, Exponential Detection Function

For the case of a continuous-effort, exponential detection function, Brown (1980) derived the following recursion and proved that it converges to the T-Optimal plan. He also showed that the solution is unique.

Let

$\xi(\cdot,t,f)$ = optimal plan for $m(t)$ effort for the stationary target distribution $q(\cdot,t,f)$.

Each step of the algorithm focuses on a single time t and computes the plan $\xi(\cdot,t,f)$. For $B = \infty$, computation of this plan may be done efficiently by the Charnes and Cooper algorithm given in Sect. 2.3.1.3 or by the algorithm given in Sect. 5.2 of Washburn (2014). If $B < \infty$, one can use the method in Sect. 2.3.1.2.

Let

$$f_t^R(\cdot,s) = \begin{cases} f(\cdot,s) & \text{for } s \neq t \\ \xi(\cdot,t,f) & \text{for } s = t. \end{cases} \tag{3.27}$$

Then $P\left(f_t^R,T\right) \geq P(f,T)$. This follows from the fact that $\xi(\cdot,t,f)$ minimizes the failure probability for the distribution $q(\cdot,t,f)$ and that

$$\begin{aligned} 1 - P(f,T) &= \sum_{j=1}^{J} e^{-W(j,t)f(j,t)} q(j,t,f) \\ &\geq \sum_{j=1}^{J} e^{-W(j,t)\xi(j,t,f)} q(j,t,f) = 1 - P\left(f_t^R,T\right). \end{aligned} \tag{3.28}$$

Brown's algorithm produces a sequence of plans with increasing detection probabilities that converge to the optimal plan as the number of steps goes to infinity. This statement is proved below. The following recursion stops after a finite number steps determined by a user-specified, small number ε

T-Optimal Search Plan Recursion
for a Continuous-Effort, Exponential Detection Function

1. Let $f_0\,(j,t) = 0$ for $j = 1,\ldots,J$ and $t = 0,\ldots,T$.
2. Let $\varepsilon > 0$ be a tolerance.
3. Set $k = 0$.
4. Set $s = k\,[\bmod\ (T+1)]$, i.e., s is the integer remainder after dividing k by $T+1$.
5. Compute $\xi\,(\cdot,s,f_k)$, the optimal plan for $q\,(\cdot,s,f_k)$ for $m(s)$ effort.
6. Set $f_{k+1}\,(\cdot,t) = \begin{cases} f_k\,(\cdot,t) & \text{for } t \neq s \\ \xi\,(\cdot,s,f_k) & \text{for } t = s. \end{cases}$
7. If $s = T$, compute $\Delta\,(f_{k+1})$. If this is less than ε stop; $P\,(f_{k+1})$ is within ε of optimal.
8. Otherwise set $k = k+1$, and go back to step 4.

Since we start the recursion with $f_0\,(j,t) = 0$ for $j = 1,\ldots,J$ and $t = 0,\ldots,T$, the first pass through the recursion from $k = 0$ to T produces f_T, the *myopic* search plan. At each time t, this plan generates the maximum increase in detection probability given the failure of the search effort at all times $s < t$. This plan is usually not T-optimal but is often close to optimal.

Since each allocation f_{k+1} increases $P\,(f_{k+1},T)$ compared to $P(f_k,T)$, it follows that $P(f_k,T)$ approaches a limit $\overline{P}$ as $k \to \infty$. Since the space of search plans $F_B(m)$ is closed and bounded, there is a subsequence $\{f_{k(n)};\ n = 1,2\ldots\}$ of the plans f_k that converges to a plan $f^* \in F_B(m)$. It follows that $P\,(f^*,T) = \overline{P}$.

The allocation f^* satisfies the necessary and sufficient conditions of Theorem 3.3. To see this we let Ξ_t be the operator that replaces a plan $f \in F(m)$ with f_t^R given in (3.27), i.e., $\Xi_t(f) = f_t^R$ and suppose that the conditions fail for some $t \in [0,T]$. Then we can strictly increase $P(f^*,T)$ by reallocating the effort at time t to obtain $\Xi_t(f^*)$ such that $P\,(\Xi_t\,(f^*)\,,T) > P\,(f^*,T) = \overline{P}$. Define $\Xi = \Xi_0 \cdots \Xi_T$. Then $P\,(\Xi\,(f^*)\,,T) > P\,(f^*,T)$. However, by the continuity of Ξ

$$P\left(\Xi\left(f^*\right),T\right) = P\left(\lim_{n\to\infty} \Xi\left(f_{k(n)}\right),T\right) = \lim_{n\to\infty} P\left(f_{k(n)+T+1},T\right) = \overline{P} \tag{3.29}$$

which is a contradiction. Therefore the conditions of Theorem 3.3 hold and f^* maximizes $P(f,T)$ for $f \in F_B(m)$.

Computing Washburn's Bound From (3.15) and the fact that the detection function is exponential, we have that

$$
\begin{aligned}
D(f,j,t) &= W(j,t)\, E_{jt}\left[\exp\left(-\sum\nolimits_{t=0}^{T} W(\omega_t,t) f(\omega_t,t)\right)\right] p_t(j) \\
&= W(j,t) \sum_{\{\omega:\omega_t=j\}} p(\omega) \exp\left(-\sum\nolimits_{t=0}^{T} W(\omega_t,t) f(\omega_t,t)\right) \\
&= W(j,t)\, e^{-W(j,t)f(j,t)} \sum_{\{\omega:\omega_t=j\}} p(\omega) \exp\left(-\sum_{s\neq t} W(\omega_s,s) f(\omega_s,s)\right) \\
&= W(j,t)\, e^{-W(j,t)f(j,t)} q(j,t,f)\,.
\end{aligned}
\tag{3.30}
$$

Computing

$$
\begin{aligned}
\underline{\lambda}(t) &= \begin{cases} \min\limits_{\{j:f(j,t)>0\}} W(j,t)\, e^{-W(j,t)f(j,t)} q(j,t,f) & \text{if } q(j,t,f) > 0 \text{ for some } j \\ 0 & \text{otherwise} \end{cases} \\
\overline{\lambda}(t) &= \begin{cases} \max\limits_{\{j:f(j,t)<B\}} W(j,t)\, e^{-W(j,t)f(j,t)} q(j,t,f) & \text{if } q(j,t,f) < B \text{ for some } j \\ \underline{\lambda}(t) & \text{otherwise,} \end{cases}
\end{aligned}
\tag{3.31}
$$

we obtain

$$
\Delta(f) = \sum\nolimits_{t=0}^{T} \left(\overline{\lambda}(t) - \underline{\lambda}(t)\right) m(t). \tag{3.32}
$$

Calculating $\Delta(f_k)$ requires one to compute $q(\cdot,t,f_k)$ for $t = 0,\ldots,T$. In the following sections, we present two implementations of Brown's recursion that provide methods of computing $q(\cdot,t,f_k)$. Experience has shown that Brown's algorithm typically produces a plan that is close to optimal after a few iterations though $[0,T]$. The reason is that the algorithm starts with the myopic plan which is often close to optimal.

3.2.2 *Implementing Brown's Recursion for an Exponential Detection Function*

The recursion below implements Brown's algorithm for an exponential detection function with the stopping rule based on Washburn's bound.

Implementation of T-Optimal Search Plan Recursion for a Continuous-Effort, Exponential Detection Function

1. Set $p^0(\omega) = \Pr\{\omega\}$ for $\omega \in \Omega$ and $f_0(j,t) = 0$ for $j = 1,\ldots,J$ and $t = 0,\ldots,T$.
2. Let $\varepsilon > 0$ be a tolerance, and set $k = 0$.
3. Set $s = k\,[\text{mod}\,(T+1)]$.

(continued)

4. Compute

$$r_k(\omega) = p^k(\omega)\exp\left(W(\omega_s, s) f_k(\omega_s, s)\right)$$
$$= p^0(\omega)\exp\left(-\sum_{t \neq s} W(\omega_t, t) f_k(\omega_t, t)\right) \text{ for } \omega \in \Omega \tag{3.33}$$

and

$$q(j, s, f_k) = \sum_{\{\omega:\omega_s = j\}} r_k(\omega) \text{ for } j = 1, \ldots, J. \tag{3.34}$$

5. Compute $\xi_k(\cdot, s)$, the optimal plan for $m(s)$ effort for $q(\cdot, s, f_k)$, and set

$$f_{k+1}(\cdot, t) = \begin{cases} f_k(\cdot, t) & \text{for } t \neq s \\ \xi_k(\cdot, s) & \text{for } t = s \end{cases}$$

and

$$p^{k+1}(\omega) = r_k(\omega)\exp\left(-W(\omega_s, s) f_{k+1}(\omega_s, s)\right) \text{ for } \omega \in \Omega. \tag{3.35}$$

6. If $s = T$, compute $\Delta(f_{k+1})$. If it is less than ε, stop.
7. Otherwise set $k = k + 1$, and return to step 3.

Note that $q(\cdot, t, f_k)$ is a defective distribution. However, by the comments in Sect. 2.4, the algorithms in Chap. 2 may be applied in step 5 to find plans for defective distributions. In the recursion, (3.34) follows from (3.22), and $\Delta(f_{k+1})$ is computed from (3.31) and (3.32). One can verify that (3.35) in step 5 produces

$$p^{k+1}(\omega) = p^0(\omega)\exp\left(-\sum\nolimits_{t=0}^{T} W(\omega_t, t) f_{k+1}(\omega_t, t)\right) \text{ for } n = 1, \ldots, N.$$

To compute $\Delta(f_{k+1})$ in step 6, we use (3.30) which requires us to compute $q(\cdot, t, f_{k+1})$ for $t = 0, \ldots, T$.

Approximating the Distribution of p on Ω The recursion given above requires almost no restrictions on the motion model other than being able to calculate the probability $p(\omega)$ of each sample path $\omega \in \Omega$. The number of possible sample paths over $[0, T]$ is $N = J^{T+1}$, so in principal, we can calculate $p(\omega)$ for all $\omega \in \Omega$. If this is impractical because the state space is too large or the distribution too complex, we can approximate the probability distribution p on Ω by simulating a large number N of sample paths from the process and replacing the original distribution with this large but finite number of sample paths.

The approach presented above is restricted to discrete time and space. However, in continuous time and space search problems, it is often convenient to impose a discrete time and space grid to approximate the continuous problem. One way to impose a discrete grid on a continuous problem is the following.

Suppose the target's motion is a modeled by a continuous space and time stochastic process $\{X(t); t \geq 0\}$. We can approximate this process by generating a large number N of sample paths from the process. This is usually done by a simulation that produces equally weighted sample paths ω each with $p(\omega) = 1/N$. These sample paths develop in continuous space and time. Next one discretizes time into instants, $s_0, s_1, \ldots$. Usually the increments $s_{i+1} - s_i$ for $i = 0, 1 \ldots$ are all equal, but this is not necessary. We impose a grid of J cells on the search space. For each simulated path ω, we compute ω_{s_i}, the cell containing the target at time s_i, for $i = 0, 1 \ldots$. Replace ω by its discretized sampled path $(\omega_{s_0}, \omega_{s_1}, \ldots)$. These N discretized sample paths produce a discrete and time and space approximation to the distribution on the set of continuous time and space paths.

Now, fix a time horizon s_T and represent the $T+1$ time instants by $t = 0, 1, \ldots, T$ where $t = i$ corresponds to time instant s_i. We employ the approximation that the target is stationary between t and $t+1$ for $t = 0, \ldots, T$. Let $m(t)$ be the search effort available in the increment $[t, t+1)$ for $t = 0, \ldots, T$. Our goal is to find a T-optimal allocation within $F(m)$. If the detection function is exponential, we can find the T-optimal plan by applying Brown's algorithm in the manner described above. The resulting allocation will be an approximation to the optimal continuous space and time plan. The quality of the approximation will depend on the number of sample paths and how fine the space and time grids are. See Chap. 5 in Shapiro et al. (2009) for a discussion of the statistical properties of solutions obtained based on Monte Carlo sampling of target paths.

Crisan (2001) shows that when the target state space S is l-dimensional Cartesian, the distribution of the state of the N sample paths at a given time t will converge to the true continuous space distribution as $N \to \infty$ in the following sense. If ν_t is the true measure at time t and ν_t^N is the measure produced by the N sample paths at time t, then for any bounded continuous function f on S,

$$E\left[\left|\int_S f(x)d\nu_t - \int_S f(x)d\nu_t^N\right|\right] \to 0 \text{ as } N \to \infty.$$

3.2.3 Example: T-Optimal and Myopic Search Plans

In this section we compare a T-optimal plan to a myopic one for a search for a boat. The example is a bit simplistic, but it is useful for illustrating the difference between these two plans.

The target, possibly a drug smuggling boat, is known to have left port at the point $(0, 0)$ at time $t = 0$ hrs. There are two possible scenarios for the motion of the target, each of which has equal weight. In scenario 1, the boat travels toward a port at $(0, 480)$ moving at approximately 20 kn. In scenario 2, the boat heads east-northeast at about 20 kn.

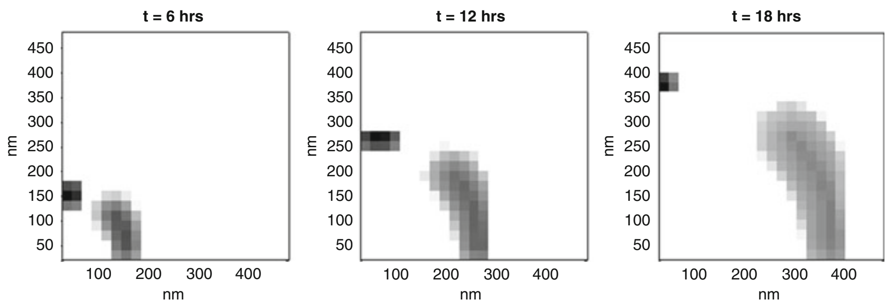


Fig. 3.1 Target location distributions – *dark cells* indicate high probability

Figure 3.1 shows the target distribution resulting from the two scenarios at the times 6, 12, and 18 h. The part of the distribution corresponding to scenario 1 shows the target heading north to the port at $(0, 480)$. In this scenario, the distribution starts at $(0, 0)$ and heads north while spreading out in the east-west direction until 12 h at which time it starts to condense and eventually ends up at $(0, 480)$ at 24 h. The part of the distribution corresponding to scenario 2 spreads out and moves in an east-northeasterly direction. To compute these distributions, we simulated 50,000 equally weighted target paths in continuous space and time, 25,000 for each scenario. At the search times we imposed a grid of cells 20 nm by 20 nm on a side. We replaced the position of each particle with the index of the cell it is in at the search times. This produced the set of paths in discrete time and space that we used for the motion model for this example. We summed the probability of the paths in each cell at the search times to produce Fig. 3.1.

We have 3000 nm^2 of search effort (swept area) available at each of the times 6, 12, and 18 h, and the detection function is exponential with

$$b(z) = 1 - e^{-z/400\ \mathrm{nm}^2} \quad \text{for} \ z \geq 0$$

for all times and cells where z is the search effort in a cell and 400 nm^2 is the area of a cell. To find the myopic and T-optimal plan for this search, we assume that effort can be distributed instantaneously over space at these times. More realistically, there will be path constraints on how the search can be applied. Finding optimal plans under searcher path constraints is the topic of Chap. 4.

Figure 3.2 shows the myopic and T-optimal search plans for $T = 24$ hrs at the three search times. To compute these plans, we used the algorithm for continuous effort and an exponential detection function given at the beginning of Sect. 3.2.2. We set $\varepsilon = 0.00001$, and in 10 iterations the Washburn bound fell below this value. This produced a detection probability 0.68 for the myopic plan and 0.76 for the T-optimal plan. Looking at Fig. 3.2, we see that the myopic plan concentrates on scenario 1 and doesn't look ahead to see that the scenario 2 distribution is spreading out. The T-optimal plan takes this into account and puts most of its effort on scenario 2 at times 6 and 12 h while waiting until 18 h to apply substantial search to scenario 1.

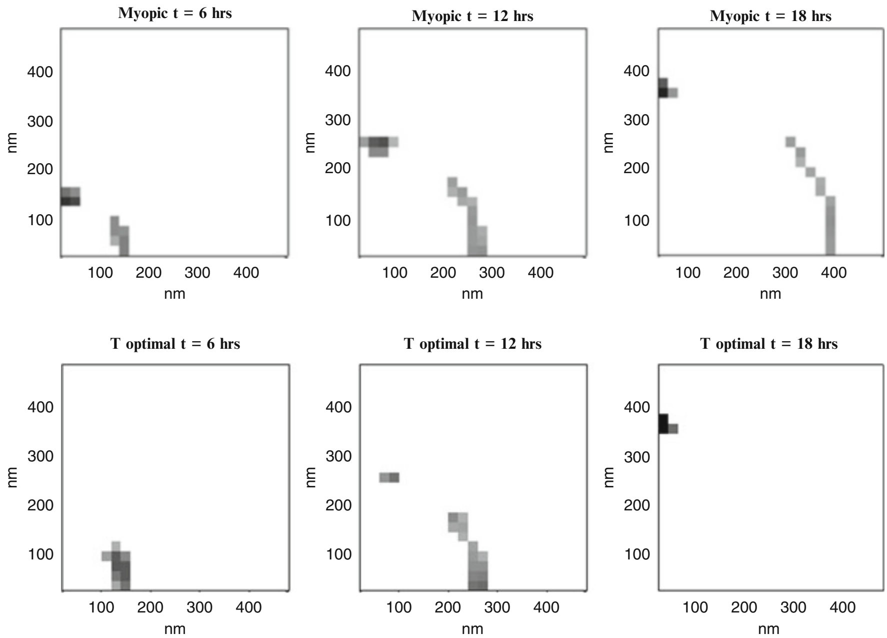


Fig. 3.2 Myopic and T-optimal search plans – *dark cells* indicate high effort

3.2.4 *Implementing Brown's Recursion for a Continuous-Effort, Exponential Detection Function and Markov Motion Model*

In this section we assume that X is a Markov process with transition function Γ. In this case,

$$\begin{aligned} p_0(j) &= \Pr\{X(0) = j\} \text{ for } j = 1, \ldots, J \text{ and} \\ \Gamma_t(i,j) &= \Pr\left\{X(t+1) = j \middle| X(t) = i\right\} \text{ for } t = 0, \ldots, T-1. \end{aligned} \tag{3.36}$$

From (3.36), we have

$$p(\omega) = \Pr\{X(t) = \omega_t,\ t = 0, \ldots, T\} = p_0(\omega_0) \prod_{t=0}^{T-1} \Gamma_t(\omega_t, \omega_{t+1}) \text{ for } \omega \in \Omega. \tag{3.37}$$

When the target motion model is Markovian, there is an efficient way to compute $q(\cdot, s, f_k)$ for step 5 in Brown's algorithm. Let

$$
\begin{aligned}
& R\,(j,t,f) \\
& = \sum_{\{\omega:\omega_t=j\}} p_0\,(\omega_0)\,\Gamma_0\Big(\omega_0,\omega_1\Big)\cdots\Gamma_{t-1}\,(\omega_{t-1},j)\exp\left(-\sum\nolimits_{s=0}^{t-1} - W\,(\omega_s,s)f\,(\omega_s,s)\right)
\end{aligned}
\tag{3.38}
$$

$$
\begin{aligned}
& S\,(j,t,f) \\
& = \sum_{\{\omega:\omega_t=j\}} \Gamma_t\Big(j,\omega_{t+1}\Big)\cdots\Gamma_{T-1}\,(\omega_{T-1},\omega_T)\exp\Big(-\sum\nolimits_{s=t+1}^{T} - W\,(\omega_s,s)f\,(\omega_s,s)\Big)\,.
\end{aligned}
\tag{3.39}
$$

Then for any allocation f,

$$
q\,(j,t,f) = R\,(j,t,f)\,S\,(j,t,f)\ \text{ for } j = 1,\ldots,J,\ t = 0,\ldots,T. \tag{3.40}
$$

An efficient algorithm for performing the T-optimal search plan recursion for continuous effort and a Markov motion model is given below. This algorithm was developed by Brown (1980).

The functions R and S are computed as part of the recursion. $R(j,t,f)$ is the probability of the target reaching state j at time t without being detected by the allocation f at the times 0 to $t-1$. Similarly, $S(j,t,f)$ is the probability of the target not being detected at times $t+1$ through T given it started in cell j at time t. Their product is the probability of the target being in cell j at time t and failing to be detected by the allocation f at any time other than t. Thus their product is equal to $q(j,t,f)$. Computation of R and S provides and efficient way of computing $q(j,t,f)$. The functions R and S are often called the reach and survive functions.

Implementation of *T*-Optimal Search Plan Recursion for a Continuous-Effort, Exponential Detection Function and Markov Motion Model

1. Let $f_0\,(j,t) = 0$ for $j = 1,\ldots,J$ and $t = 0,\ldots,T$.
2. Set

$$
\begin{aligned}
& S\,(j,t,f_0) = 1 \text{ for } j = 1,\ldots,J \text{ and } t = 0,\ldots,T \\
& R\,(j,0,f_0) = p_0(j) \text{ for } j = 1,\ldots,J.
\end{aligned}
$$

3. Set $k = 0$, and let $\varepsilon > 0$ be tolerance.
4. Set $s = k\,[\bmod\ (T+1)]$.
5. Compute

$$
\begin{aligned}
& q\,(j,s,f_k) = R\,(j,s,f_k)\,S\,(j,s,f_k)\ \text{ for } j = 1,\ldots,J \text{ and} \\
& \xi_k\,(\cdot,s)\,,\ \text{the optimal plan for } q\,(\cdot,s,f_k)\,.
\end{aligned}
$$

(continued)

6. Set

$$f_{k+1}(\cdot, t) = \begin{cases} f_k(\cdot, t) & \text{for } t \neq s \\ \xi_k(\cdot, s) & \text{for } t = s. \end{cases}$$

7. If $s < T$, compute

$$R(j, s+1, f_{k+1}) = \sum_{i=1}^{J} R(i, s, f_k)\, e^{-W(i,s)\xi(i,s,f_k)} \Gamma_s(i, j) \quad \text{for } j = 1, \ldots, J,$$

   set $k = k + 1$, and return to step 4.

8. If $s = T$, compute $\Delta(f_{k+1})$. If it is less than ε stop.
9. If $s = T$, and $\Delta(f_{k+1}) > \varepsilon$, set

$$S(j, T, f_{k+1}) = 1 \text{ and } R(j, 0, f_{k+1}) = p_0(j) \text{ for } j = 1, \ldots, J,$$

and for $t = T - 1, \ldots, 0$, and $j = 1, \ldots, J$, compute

$$S(j, t, f_{k+1}) = \sum\nolimits_{i=1}^{J} \Gamma_t(j, i)\, e^{-W(i,t+1)f_{k+1}(i,t+1)} S(i, t+1, f_{k+1})$$

10. Set $k = k + 1$ and return to step 4.

3.2.5 Extensions of the Optimal Detection Problem

The results obtained above can be extended in a number of ways.

Optimal Survivor Search In some cases, particularly for problems involving people lost at sea or in a wilderness area, maximizing probability of detection is not the main goal. Instead, the goal is to detect the person while they are still alive. This problem can be handled by a straight-forward extension of the optimal detection search. One adds a $J + 1^{st}$ state, person dead, to the target state space and sets $W(J + 1, t) = 0$ for $t = 0, \ldots, T$. The target motion model is extended to consider the possibility that the search object may die (transition to state $J + 1$) as time and exposure increases. This is particularly critical for people in the water. When the water is cold, hypothermia can overcome a person quickly. There are tables developed by the Coast Guard that give the probability of dying as function time and water temperature. When this possibility is included in the "motion" model for the search object and the sweep width for state $J + 1$ set to 0, then maximizing the probability of detection by time T is equivalent to maximizing the probability of finding the person alive by time T and the methods of this section apply to finding the optimal survivor search plan.

Optimal Defensive Search In optimal defensive search, one is trying to detect the target before it reaches a certain state. The state might represent launching an attack. The solution to this problem is similar to that for the survivor search problem. Add or designate a set of states that you are trying to prevent the target from reaching. Model the target's motion so that the states in this set are trapping states. Once the target enters a trapping state, it stays there. Set the sweep equal to 0 in these states. The plan that maximizes the probability of detection by time T also maximizes the probability of detecting the target before it enters one of the trapping states.

Optimal Whereabouts Search In a whereabouts search, one succeeds by detecting the target by time T, or if the target is not detected, then the searcher is allowed to guess one of the cells in the whereabouts grid. If the target is in the guessed cell, the search succeeds. Each cell in the whereabouts grid consists of a set of cells in the target state space. The set of target state cells in one whereabouts grid cell does not overlap with the set of target state cells in any other whereabouts grid cell.

Supposed there are N_W whereabouts cells. The solution to the optimal whereabouts search is computed by solving N_W optimal detection search problems as follows. For $n = 1, \ldots, N_W$, let $S(n)$ be the set of target state cells that comprise the nth whereabouts cell. A whereabouts plan consists of a pair (f, n) where $f \in F_B(m)$ and $1 \le n \le N_W$. The plan proceeds by performing the search f and if that fails to detect the target, guessing the whereabouts cell $S(n)$. Let

$$P^n(f, T) = \text{probability of detecting target by time } T \text{ with plan } f \text{ given the target is not in } S(n) \text{ at time } T.$$

Then the probability of success for plan (f, n) is

$$P_W(f, T, n) = P^n(f, T)\,(1 - \Pr\{X(T) \in S(n)\}) + \Pr\{X(T) \in S(n)\}. \tag{3.41}$$

Given we have decided to choose the nth whereabouts cell, the plan $f^{(n)} \in F_B(m)$ that maximizes $P^n(f, T)$ will maximize $P_W(f, T, n)$ for $f \in F_B(m)$. Thus we find the optimal whereabouts plan by finding the $n*$ such that

$$P_W\left(f^{(n*)}, T, n*\right) = \max_{1 \le n \le N_W} P_W\left(f^{(n)}, T, n\right).$$

To find $f^{(n)}$ we must construct the target motion process $X^{(n)} = \left\{X \middle| X(T) \notin S(n)\right\}$. In the general discrete space and time case where the motion process is defined in terms of sample path probabilities as in (3.1), the process $X^{(n)}$ is obtained by deleting all samples paths for which $X(T) \in S(n)$ and renormalizing the probabilities on the remaining sample paths to add to 1. If the detection function is exponential, then we can find $f^{(n)}$ by the algorithm in Sect. 3.2.2. If the target motion process is Markovian, then we can construct $X^{(n)}$ by setting

$$\Gamma_{T-1}^{(n)}(i,j) = \begin{cases} \gamma^{(i)}\Gamma_{T-1}(i,j) & \text{for } j \notin S(n) \\ 0 & \text{for } j \in S(n) \end{cases} \quad \text{for } i = 1, \ldots, J$$

where $\gamma^{(i)}$ is a positive constant such that

$$\gamma^{(i)} \sum_{j \notin S(n)} \Gamma_{T-1}^{(n)}(i,j) = 1 \text{ for } i = 1, \ldots, J.$$

Replacing Γ_{T-1} by $\Gamma_{T-1}^{(n)}$ in (3.36) yields the process $X^{(n)}$ and the plan $f^{(n)}$ can be found using the algorithm in Sect. 3.2.4.

3.3 Discrete-Effort Search Problems

The model for discrete-effort search problems is the same as for the continuous-effort problems defined as in Sect. 3.1 with the exception that search effort must be allocated in discrete looks.

Discrete-Effort Search Plan A discrete-effort search plan is a space-time search allocation f where $f(j,t)$ specifies the number of looks in cell j at time t.

We assume that $m(t) \geq 0$ looks are available for $t \geq 0$. We define the class $F_d(m)$ of discrete-effort search plans as follows:

$$\begin{aligned} & f \in F_d(m) \text{ if and only if} \\ & f(j,t) \in \{0,1,\ldots\} \text{ for } 1 \leq j \leq J \text{ and } t = 0,\ldots,T \\ & \textstyle\sum_{j=1}^{J} f(j,t) = m(t) \text{ for } t = 0,\ldots,T. \end{aligned}$$

The other definitions in Sect. 3.1 remain the same for discrete-effort searches except that the class of plans $F(m)$ is replaced by $F_d(m)$.

3.3.1 Necessary Conditions for a T-Optimal Discrete-Effort Plan When the Detection Function Is Exponential

In this section, we derive the analog of the necessary conditions given in Theorem 3.3 for the case where the discrete effort detection function is exponential. Because the set of plans $F_d(m)$ is not convex, these conditions are not sufficient.

When the detection function is exponential, the probability of detecting the target with allocation $f \in F_d(m)$ by time t given it follows path ω is

$$b(\zeta(f,\omega,t)) = 1 - \exp\left(-\textstyle\sum_{s=0}^{t} W(\omega_s,s) f(\omega_s,s)\right). \tag{3.42}$$

Let

$$\beta(j,s) = 1 - e^{-W(j,s)} \text{ for } s = 0,\ldots,T,\ j = 1,\ldots,J.$$

Equation (3.42) is equivalent to assuming that, given the target is in cell j at time s, a look in that cell has an independent opportunity to detect with probability $\beta(j,s)$. Obviously, if one starts by specifying $\beta(j,s)$, then one can obtain the detection function in (3.42) by setting $W(s,j) = -\ln(1-\beta(j,s))$ for $s = 0,\ldots,T$ and $j = 1,\ldots,J$.

We now state the analog of the necessary conditions of Theorem 3.3 for discrete-effort, exponential detection functions.

Theorem 3.4 *Assume b is a discrete-effort, exponential detection function. Then a necessary condition for $f^* \in F_d(m)$ to be a T-optimal plan is that $f^*(\cdot,t)$ be the optimal stationary target plan for cost $m(t)$ for the distribution $\tilde{q}(\cdot,t,f^*)$, which is the posterior probability distribution on the target's location at time t given failure to detect at all times other than t, for $t = 0,\ldots,T$.*

Proof Suppose that $f^* \in F_d(m)$ is a T-optimal plan. From (3.23), we have

$$1 - P(f^*,T) = \sum_{j=1}^{J} e^{-W(j,t)f^*(j,t)} q(j,t,f^*) \text{ for } t = 0,\ldots,T.$$

Since $f^* \in F_d(m)$ is T-optimal, it minimizes $1 - P(f^*,T)$, and $f^*(\cdot,t)$ must minimize the probability of failure for the stationary target search with probability distribution $q(\cdot,t,f^*)$ among plans with $m(t)$ looks for $t = 0,\ldots,T$. Thus $f^*(\cdot,t)$ is the optimal stationary target plan for cost $m(t)$ for the distribution $\tilde{q}(\cdot,t,f^*)$ for $t = 0,\ldots,T$. This proves the theorem. This result was obtained by Washburn (1980)

One can employ the recursion given below (3.28) to generate a sequence of discrete-effort plans and continue this recursion until no change in the plan occurs during a complete cycle through $[0,T]$. The resulting plan will satisfy the necessary conditions of Theorem 3.4, but the plan may not be optimal because the recursion may have found a local rather than a global optimum. To try to overcome this, one can employ several different starting allocations in addition to the one that produces the myopic plan for the first pass. However, to be guaranteed of obtaining an optimal plan, one must use a nonlinear integer program solver.

3.3.2 Nonlinear Integer Programming Formulation

We can formulate the problem of finding a T-optimal plan as a nonlinear integer program. There are nonlinear, integer-programming solvers that are capable of solving large integer programs, so this may be a viable approach for some problems. Finding a T-optimal plan is equivalent to solving the following nonlinear, integer program.

$$\begin{aligned}&\textbf{Find } f \text{ to}\\&\text{maximize } P(f,T)=\sum_{\omega\in\Omega}p(\omega)\,b\left(\sum\nolimits_{t=0}^{T}W(\omega_t,t)f(\omega_t,t)\right)\end{aligned} \tag{3.43}$$

$$\begin{aligned}&\textbf{Subject to} \text{ the following constraints}\\&\quad f(j,t)\in\{0,1,\ldots\}\ \text{for } j=1,\ldots,J \text{ and } t=0,\ldots,T\\&\quad \sum\nolimits_{j=1}^{J}f(j,t)=m(t)\ \text{for } t=0,\ldots,T.\end{aligned} \tag{3.44}$$

If b is exponential, then the above program is equivalent to

$$\begin{aligned}&\textbf{Find } f \text{ to}\\&\text{minimize } \sum_{\omega\in\Omega}p(\omega)\exp\left(-\sum\nolimits_{t=0}^{T}W(\omega_t,t)f(\omega_t,t)\right)\end{aligned} \tag{3.45}$$

Subject to the constraints in (3.44).

If the number of sample paths in Ω is large, it may not be practical to solve the above program. In this case one can replace Ω with a sample of N paths chosen in a fashion similar to that discussed in Sect. 3.2.2. The number N must be chosen large enough to provide a good representation of the target motion process but small enough to allow for solution of the resulting program.

3.4 Forward and Backward Algorithm

This section generalizes Brown's recursion from Sect. 3.2 to the Forward And Backward (FAB) algorithm developed by Washburn (1983). The FAB algorithm applies to more general payoff functions than probability of detection. These more general functions involve linear combinations of detection functions.

As an example, suppose we wish to minimize mean time to complete a search. A search completes when the target is detected or at time $T+1$ if the target is not detected. The mean time to completion is not the mean time to detection since for completion we quit at time T if we have not detected the target. We can approximate minimizing the mean time to detection by taking T to be large.

For a search plan let $M(f)$ be the mean time to completion using this plan. Since

$$P(f,t)=\sum_{\omega\in\Omega}p(\omega)\,b(\zeta(\omega,t,f)),$$

one can verify that

$$\begin{aligned} M(f) &= \sum_{t=1}^{T} t\,[P\,(f,t) - P\,(f,t-1)] + (T+1)\,[1 - P\,(f,T)] \\ &= T + 1 + \sum_{t=0}^{T} [tP\,(f,t) - (t+1)\,P\,(f,t)] \\ &= \sum_{t=0}^{T} (1 - P\,(f,t)) \end{aligned} \tag{3.46}$$

and that (3.46) holds even when $T = \infty$. It follows that

$$-M(f) = -\sum_{\omega\in\Omega} p\,(\omega) \sum_{t=0}^{T} [1 - b\,(\zeta\,(\omega,t,f))]. \tag{3.47}$$

For a continuous-effort, exponential detection function, the FAB algorithm, described below, finds a plan to minimize $M(f)$ by finding f to maximize $-M(f)$.

FAB Payoff Functions For $f \in F$, define

$$\begin{aligned} &L\,(f,\omega) = a_{T+1}\,(\omega) + \sum_{t=0}^{T} a_t\,(\omega)\,b\,(\zeta\,(\omega,t,f)) \text{ for } \omega \in \Omega\,, \\ &\widehat{P}(f) = E\,[L\,(f,\omega)]\,. \end{aligned} \tag{3.48}$$

We call payoff functions that are in the form (3.48), *FAB payoff functions.* The coefficients a_t for $t = 0, \ldots, T+1$, are functions of the target's sample path. As an example they can depend on the target state at time t, or they can depend on the whole path of the target. They can also be constants.

If we set $a_{T+1} = -(T+1)$ and $a_t = 1$ for $t = 0, \ldots, T$, then $\widehat{P}(f)$ becomes the negative of the mean time to complete a search with plan f. If we set $a_t = 0$ for $t = 0, \ldots, T-1$, $a_T = 1$, and $a_{T+1} = 0$, then $\widehat{P}(f)$ becomes the probability of detecting the target by time T.

Suppose that are our search is limited to $[0, T]$ and we receive a reward $\widehat{r}(t)$ if we detect the target at time t. If we don't detect by time T, we obtain reward $\widehat{r}\,(T+1)$. If $\widehat{r}$ is a decreasing function of time, we show, at the end of this section that the expected reward obtained by a search plan is a FAB payoff function. This expected reward function is obtained by setting

$$\begin{aligned} &a_t(t) = \widehat{r}(t) - \widehat{r}\,(t+1) \;\text{ for } t = 0, \ldots, T, \text{ and} \\ &a_{T+1} = \widehat{r}\,(T+1)\,. \end{aligned}$$

Theorem 3.5: Discrete Space and Time Optimality Conditions for FAB Payoff Functions *Assume b is a continuous-effort, decreasing-rate detection function. Let* $L(f,\omega)$ *defined in* (3.48) *be a concave function of f where* $a_t, t = 0, \ldots, T$ *are non-negative. By the Kuhn-Tucker-Karush Theorem, a necessary and sufficient condition for* $f^* \in F_B(m)$ *to satisfy*

$$\widehat{P}\left(f^*\right) \geq \widehat{P}(f) \;\text{ for }\; f \in F_B(m)$$

is the existence of a vector $(\lambda_0, \ldots, \lambda_T)$ *with positive components such that for* $t = 0, \ldots, T$

$$\frac{\partial \widehat{P}(f^*)}{\partial f^*(j,t)} \begin{array}{l} \le \lambda_t \text{ if } f^*(j,t) = B \\ = \lambda_t \text{ if } 0 < f^*(j,t) < B \\ \le \lambda_t \text{ if } f^*(j,t) = 0. \end{array} \text{ for } j = 1, \ldots, J. \tag{3.49}$$

Calculating $\partial \widehat{P}(f)/\partial f(j,t)$. Since

$$\widehat{P}(f) = E\left[L(b,f,\omega)\right] = \sum_{\omega \in \Omega} p(\omega)\, L(b,f,\omega), \tag{3.50}$$

we can write $\partial \widehat{P}(f)/\partial f(j,t)$ more explicitly as follows.

$$\begin{aligned} \frac{\partial \widehat{P}(f)}{\partial f(j,t)} &= \sum_{\omega \in \Omega} p(\omega) \frac{\partial L(b,f,\omega)}{\partial f(j,t)} \\ &= \sum_{\omega \in \Omega} p(\omega) \sum_{s=0}^{T} a_s(\omega)\, b'(\zeta(\omega,s,f)) \frac{\partial \zeta(\omega,s,f)}{\partial f(j,t)} \\ &= \sum_{\{\omega:\omega_t = j\}} p(\omega) \sum_{s=t}^{T} a_s(\omega)\, b'(\zeta(\omega,s,f))\, W(j,t). \end{aligned} \tag{3.51}$$

The summation in the last line of (3.51) excludes paths for which $\omega_t \neq j$ because those terms do not depend on $f(j,t)$. By the same reasoning, the terms in the summation corresponding to times $s < t$ are equal to 0.

Define $p_t(j) = \Pr\{X(t) = j\}$, and let E_{jt} indicate expectation conditioned on $X(t) = j$. We can write (3.51) in the form

$$\frac{\partial \widehat{P}(f)}{\partial f(j,t)} = W(j,t)\, E_{jt}\left[\sum_{s=t}^{T} a_s(\omega)\, b'(\zeta(\omega,s,f))\right] p_t(j). \tag{3.52}$$

3.4.1 Bound on FAB Payoff Function for Continuous-Effort, Decreasing-Rate Detection Functions

We generalize the Washburn bound obtained in Sect. 3.1.2 to FAB payoff functions. Recall that if b is a continuous-effort, decreasing-rate detection function then

$$b(z_2) - b(z_1) \le b'(z_1)(z_2 - z_1).$$

Suppose $f_1, f_2 \in F(m)$, then

$$\begin{aligned}
&\widehat{P}(f_2) - \widehat{P}(f_1) \\
&\quad = E\left[\sum\nolimits_{t=0}^{T} a_t(\omega)\, b(\zeta(\omega,t,f_2))\right] - E\left[\sum\nolimits_{t=0}^{T} a_t(\omega)\, b(\zeta(\omega,t,f_1))\right] \\
&\quad \le E\left[\sum\nolimits_{t=0}^{T} a_t(\omega)\, b'(\zeta(\omega,t,f_1)) \sum\nolimits_{s=0}^{t} W(\omega_s,s)\,(f_2(\omega_s,s) - f_1(\omega_s,s))\right].
\end{aligned} \tag{3.53}$$

From (3.53), we have

$$\begin{aligned}
&\widehat{P}(f_2) - \widehat{P}(f_1) \\
&\le E\left[\sum_{t=0}^{T}\sum_{s=0}^{t} a_t(\omega_t)\, b'(\zeta(\omega,t,f_1)) W(\omega_s,s)\,(f_2(\omega_s,s) - f_1(\omega_s,s))\right] \\
&= E\left[\sum_{s=0}^{T}\sum_{t=s}^{T} a_t(\omega_t)\, b'(\zeta(\omega,t,f_1)) W(\omega_s,s)\,(f_2(\omega_s,s) - f_1(\omega_s,s))\right] \\
&= \sum_{s=0}^{T} E\left[W(\omega_s,s) \sum_{t=s}^{T} a_t(\omega_t)\, b'(\zeta(\omega,t,f_1))\,(f_2(\omega_s,s) - f_1(\omega_s,s))\right].
\end{aligned} \tag{3.54}$$

We can expand the last line of (3.54) to obtain

$$\begin{aligned}
&\widehat{P}(f_2) - \widehat{P}(f_1) \\
&\le \sum_{s=0}^{T}\sum_{j=1}^{J}\sum_{\{\omega:\omega_s=j\}} W(j,s)\, E_{js}\left[\sum_{t=s}^{T} a_t(\omega_t)\, b'(\zeta(\omega,t,f_1))\,(f_2(j,s) - f_1(j,s))\right] \\
&= \sum\nolimits_{s=0}^{T}\sum\nolimits_{j=1}^{J} D(f_1,j,s)\,(f_2(j,s) - f_1(j,s))
\end{aligned} \tag{3.55}$$

where

$$\begin{aligned}
D(f,j,t) &= W(j,t)\, E_{jt}\left[\sum_{s=t}^{T} a_s(\omega_s) b'(\zeta(\omega,s,f))\right] p_t(j) \\
&= W(j,t) \sum_{\{\omega:\omega_t=j\}} p(\omega) \sum\nolimits_{s=t}^{T} a_s(\omega_s)\, b'(\zeta(\omega,s,f)).
\end{aligned} \tag{3.56}$$

Defining $\overline{\lambda}(f_1,t)$ and $\underline{\lambda}(f_1,t)$ as in (3.17), we have by (3.55) and the proof of Theorem 3.2

$$\widehat{P}(f_2) - \widehat{P}(f_1) \le \sum\nolimits_{t=0}^{T} \left(\overline{\lambda}(f_1,t) - \underline{\lambda}(f_1,t)\right) m(t) \equiv \Delta(f_1)\,. \tag{3.57}$$

so that for f^* that maximizes $\widehat{P}(f)$ over $f \in F_B(m)$, we have

$$\widehat{P}\left(f^*\right) \le \widehat{P}(f) + \Delta(f) \text{ for any } f \in F_B(m) \tag{3.58}$$

which provides a useful upper bound on $\widehat{P}(f^*)$.

Exponential Detection Function Suppose $b(z) = 1 - e^{-z}$ for $z \geq 0$. For $f \in F$, let

$$\alpha\left(f, \omega, r\right) = a_r\left(\omega\right) e^{-\zeta(\omega, r, f)} \text{ for } \omega \in \Omega,\ r = 0, \ldots, T. \tag{3.59}$$

For $j = 1, \ldots, J,\ s = 0, \ldots, T$, define

$$d\left(f, j, s\right) = e^{W(j,s)f(j,s)} \sum_{\{\omega:\omega_s=j\}} p\left(\omega\right) \sum_{r=s}^{T} \alpha\left(f, \omega, r\right). \tag{3.60}$$

Let $A = \sum_{t=0}^{T+1} E\left[a_t\left(\omega\right)\right]$. Then for $f \in F$,

$$A - \widehat{P}(f) = \sum_{j=1}^{J} e^{-W(j,s)f(j,s)} d\left(f, j, s\right). \tag{3.61}$$

From (3.56) and (3.51), we have

$$D\left(f, j, s\right) = \frac{\partial \widehat{P}(f)}{\partial f\left(j, s\right)} = W\left(j, s\right) e^{-W(j,s)f(j,s)} d\left(f, j, s\right). \tag{3.62}$$

3.4.2 FAB Algorithm for an Exponential Detection Function

Below we state the FAB recursion for a continuous-effort, exponential detection function. We assume that a_t is non-negative for $t = 0, \ldots, T+1$ so that $\widehat{P}(f)$ in (3.48) is a concave function of $f \in F$. This ensures that the assumptions of Theorem 3.5 are satisfied.

FAB Algorithm
for a Continuous-Effort, Exponential Detection Function

1. Let $f_0\left(j, t\right) = 0$ for $j = 1, \ldots, J$ and $t = 0, \ldots, T$.
2. Let $\varepsilon > 0$ be small number, and set $k = s = 0$.
3. For $r = 0, \ldots, T$, set

$$\alpha\left(f_0, \omega, r\right) = a_r\left(\omega\right) \text{ for } \omega \in \Omega$$
$$d\left(f_0, j, r\right) = \sum_{\{\omega:\omega_r=j\}} p\left(\omega\right) \sum_{u=r}^{T} \alpha\left(f_0, \omega, u\right) \text{ for } j = 1, \ldots, J.$$

4. Find $\xi_k\left(\cdot, s\right)$ to minimize

$$\sum_{j=1}^{J} e^{-W(j,s)\xi_k(j,s)} d\left(f_k, j, s\right) \tag{3.63}$$

subject to $\xi_k\left(j, s\right) \geq 0$ for $j = 1, \ldots, J$, and $\sum_{j=1}^{J} \xi_k\left(j, s\right) = m(s)$.

(continued)

5. Set

$$f_{k+1}(\cdot,t) = \begin{cases} f_k(\cdot,t) & \text{for } t \neq s \\ \xi_k(\cdot,s) & \text{for } t = s. \end{cases}$$

6. If $s = T$, compute $\Delta(f_{k+1})$. If this is less than ε stop.
7. Otherwise, for $\omega \in \Omega$, set

$$\alpha(f_{k+1},\omega,r) = \begin{cases} e^{W(\omega_s,s)(f_k(\omega_s,s)-\xi_k(\omega_s,s))}\alpha(f_k,\omega,r) & \text{for } r \geq s \\ \alpha(f_k,\omega,r) & \text{for } 0 \leq r < s. \end{cases} \tag{3.64}$$

8. Then set $s^+ = s + 1\,[\bmod (T+1)]$, and compute

$$d\left(f_{k+1},j,s^+\right) = e^{W\left(j,s^+\right)f_{k+1}\left(j,s^+\right)} \sum_{\{\omega:\omega_{s^+}=j\}} p(\omega) \sum_{r=s^+}^{T} \alpha(f_{k+1},\omega,r) \quad \text{for } j = 1,\ldots,J.$$

9. Set $k = k+1$, $s = s^+$ and go back to step 4.

In step 6, we compute $\Delta(f_{k+1})$ by the use of (3.62). Step 7 of the algorithm removes the effect of the old allocation $f_k(\cdot,s)$ at time s and accounts for the effect of the new allocation $\xi_k(\cdot,s)$ on $\alpha(f_k,\omega,r)$ for $r \geq s$ to obtain $\alpha(f_{k+1},\omega,r)$.

For $B = \infty$, we can accomplish step 4, by normalizing $d(f_k,\cdot,s)$ to a probability distribution and finding $\xi_k(\cdot,s)$ by using the Charnes and Cooper algorithm given in Sect. 2.3.1.3 or the algorithm given in Sect. 5.2 of Washburn (2014). If $B < \infty$, we can use the algorithm in Sect. 2.3.1.2.

In step 6, the algorithm checks Washburn's bound after each pass through the times from 0 to T to see if the stopping criterion has been reached. If it has, the plan f_{k+1} is within ε of the optimal plan. The bound is easily computed using (3.62).

From (3.61) we see that each allocation f_{k+1} increases the payoff $\widehat{P}(f_{k+1})$ compared to $\widehat{P}(f_k)$. It follows that $\widehat{P}(f_k)$ approaches a limit $\overline{P}$ as $k \to \infty$. Since the space of search plans $F_B(m)$ is compact, there is a subsequence of plans that converges to a plan $f^* \in F_B(m)$ and $\widehat{P}(f^*) = \overline{P}$.

It follows that for each $s = 0,\ldots,T$ there is a $\lambda_s > 0$ such that

$$\frac{\partial\widehat{P}(f^*)}{\partial f(j,s)} \begin{array}{l} \geq \lambda_s \text{ if } f^*(j,s) = B \\ = \lambda_s \text{ if } 0 < f^*(j,s) < B \\ \leq \lambda_s \text{ if } f^*(j,s) = 0 \end{array} \quad \textit{for } j = 1,\ldots,J. \tag{3.65}$$

To see this we suppose that (3.65) does not hold for some time s. Then we could strictly increase $\widehat{P}(f^*)$ by reallocating the effort at time s as in step 4 to obtain

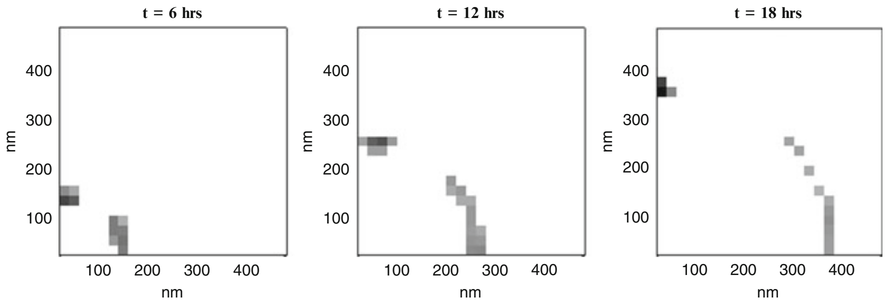


Fig. 3.3 Plan that minimizes mean time to completion – *dark cells* indicate high effort

$f_s^R \in F_B(m)$ such that $\widehat{P}\left(f_s^R\right) > \widehat{P}(f^*) = \overline{P}$. However, an argument similar to that given in Sect. 3.2.1 shows this is not possible. Therefore (3.65) holds for $s = 0, \ldots, T$, and the necessary and sufficient conditions of Theorem 3.1 are satisfied which means f^* maximizes $\widehat{P}(f)$ for $f \in F_B(m)$.

3.4.3 Example: Minimizing Mean Time to Completion

In this example, we apply the FAB algorithm to find the plan that minimizes mean time to completion for the search scenario in the example in Sect. 3.2.3.

For the problem described in Sect. 3.2.3, we applied the FAB algorithm for continuous effort and exponential detection function given above. We set $\varepsilon = 0.00001$ and iterated until the Washburn bound fell below ε. This required 8 iterations. Figure 3.3 shows the resulting plan which is similar to the myopic plan in Fig. 3.2. The mean time for the optimal completion plan is 13.47 h compared to 13.61 h for the myopic plan and 15.92 h for the T-optimal plan. By contrast, the probability of detection for the optimal completion plan is 0.71 compared to 0.76 for the T-optimal plan and 0.68 for the myopic plan. In this example, the optimal completion plan appears to be a compromise between detecting the target as early as possible and reaching the highest detection probability by time T.

3.4.4 FAB Algorithm for Continuous-Effort, Exponential Detection Function and Markov Target Motion

This section presents the special case of the FAB algorithm that was developed by Washburn (1983) for Markov target motion.

As in Sect. 3.2.4, we assume that X is a Markov process with transition function Γ. Specifically,

$$\begin{aligned} &p_0(j) = \Pr\{X(0) = j\} \text{ for } j = 1, \ldots, J \text{ and} \\ &\Gamma_t(i,j) = \Pr\left\{X(t+1) = j \middle| X(t) = i\right\} \text{ for } t = 0, \ldots, T-1. \end{aligned} \tag{3.66}$$

We make the additional assumption that $a_t(\omega)$ does not depend on ω_s for $s < t$ for $t = 0, \ldots, T$.

We now define, in a recursive fashion, functions R and S which are the analogs of the reach and survive functions defined for the Markov target motion version of Brown's algorithm. For $f \in F$ and $j = 1, \ldots, J$, let

$$\begin{aligned} &R(j,T,f) = E_{jT}[a_T(\omega)] \\ &\text{and for } t = T-1, \ldots, 0 \\ &R(j,t,f) = E_{jt}[a_t(\omega)] + \sum_{i=1}^{J} \Gamma_t(j,i)\, e^{-W(i,t+1)f(i,t+1)} R(i,t+1,f). \end{aligned} \tag{3.67}$$

For $f \in F$ and $j = 1, \ldots, J$, let

$$\begin{aligned} &S(j,0,f) = p_0(j) \\ &\text{and for } t = 1, \ldots, T \\ &S(j,t,f) = \sum_{i=1}^{J} \Gamma_{t-1}(i,j)\, e^{-W(i,t-1)f(i,t-1)} S(i,t-1,f). \end{aligned} \tag{3.68}$$

From the above definitions and the assumption that for $t = 0, \ldots, T$, $a_t(\omega)$ does not depend on ω_s for $s < t$, we have

$$\begin{aligned} R(j,t,f) &= E_{jt}\left[\sum_{s=t}^{T} a_s(\omega)\, e^{-\sum_{r=t+1}^{s} W(\omega_r, r) f(\omega_r, r)}\right] \\ S(j,t,f) &= E_{jt}\left[e^{-\zeta(\omega,t-1,f)}\right] p_t(j). \end{aligned}$$

Let $A = \sum_{t=0}^{T+1} E[a_t(\omega)]$. Observe that for any $t = 0, \ldots, T$,

$$A - \widehat{P}(f) = E\left[\sum_{s=0}^{t-1} a_s(\omega)\, e^{-\zeta(\omega,s,f)}\right] + E\left[\sum_{s=t}^{T} a_s(\omega)\, e^{-\zeta(\omega,s,f)}\right] \tag{3.69}$$

and that the first term on the right-hand side of (3.69) does not depend on $f(\cdot, t)$. We can write the second term as

$$
\begin{aligned}
&E\left[\sum_{s=t}^{T} a_s(\omega)\, e^{-\zeta(\omega,s,f)\big)}\right] \\
&= \sum_{j=1}^{J} p_t(j) E_{jt}\left[\sum_{s=t}^{T} a_s(\omega)\, e^{-\zeta(\omega,s,f)}\right] \\
&= \sum_{j=1}^{J} e^{-W(j,t)f(j,t)} p_t(j) E_{jt}\left[\sum_{s=t}^{T} a_s(\omega)\, e^{-\sum\limits_{r=0}^{t-1} W(\omega,r) f(\omega,r)}\, e^{-\sum\limits_{r=t+1}^{s} W(\omega,r) f(\omega,r)}\right].
\end{aligned}
$$

By the Markov property,

$$
\begin{aligned}
&E\left[\sum_{s=t}^{T} a_s(\omega)\, e^{-\zeta(\omega,s,f)\big)}\right] \\
&= \sum_{j=1}^{J} e^{-W(j,t)f(j,t)} p_t(j) E_{jt}\left[e^{-\zeta(\omega,t-1,f)}\right] E_{jt}\left[\sum_{s=t}^{T} a_s(\omega_s)\, e^{-\sum\limits_{r=t+1}^{s} W(\omega,r) f(\omega,r)}\right] \\
&= \sum_{j=1}^{J} e^{-W(j,t)f(j,t)} S(j,t,f) R(j,t,f)\,.
\end{aligned}
\tag{3.70}
$$

We now state the FAB algorithm for Markovian target motion.

FAB Algorithm for Markovian Target Motion and a Continuous-Effort, Exponential Detection Function

1. Let $f_0(j,t) = 0$ for $j = 1, \ldots, J$ and $t = 0, \ldots, T$.
2. Let $\varepsilon > 0$ be small number, set $k = s = 0$, and let

$$S(j,0,f) = p_0(j) \ \text{ for } \ j = 1,\ldots,J \text{ and } f \in F.$$

3. For $f = f_k$ and $j = 1, \ldots, J$, compute

$$R(j,T,f) = E_{jT}\left[a_T(\omega)\right],$$

and for $t = T-1, \ldots 0$,

$$R(j,t,f) = E_{jt}\left[a_t(\omega)\right] + \sum_{i=1}^{J} \Gamma_t(j,i)\, e^{-W(i.t+1)f(i,t+1)} R(i,t+1,f).$$

4. Find $\xi_k(\cdot, s)$ to minimize

(continued)

$$\sum_{j=1}^{J} e^{-W(j,s)\xi_k(j,s)} S(j,s) R(j,s)$$

subject to $\xi_k(j,s) \geq 0$ for $j = 1, \ldots, J,$ and $\sum_{j=1}^{J} \xi_k(j,s) = m(s)$.

5. Set

$$f_{k+1}(\cdot,t) = \begin{cases} f_k(\cdot,t) & \text{for } t \neq s \\ \xi_k(\cdot,s) & \text{for } t = s. \end{cases}$$

6. If $s = T$, compute $\Delta(f_{k+1})$. If this is less than ε stop. If $\Delta(f_{k+1}) > \varepsilon$, set $s = 0,\ k = k + 1$, and go to step 3.
7. Otherwise set $s = s + 1$, $k = k + 1$, compute

$$S(j,s,f_k) = \sum_{i=1}^{J} \Gamma_{t-1}(i,j)\, e^{-W(i,s-1)f_k(i,s-1)} S(i,s-1,f_k) \quad \text{for } j = 1, \ldots, J,$$

and go to step 4.

By virtue of (3.69) and (3.70), the allocation f_{k+1} found in steps 4 and 5 has the property that

$$A - \widehat{P}(f_{k+1}) \leq A - \widehat{P}(f_k) \tag{3.71}$$

so that $\widehat{P}(f_{k+1}) \geq \widehat{P}(f_k)$ for $k = 0, 1, \ldots,$ and

$$\lim_{k \to \infty} \widehat{P}(f_k) = \widehat{P}(f^*)$$

where f^* maximizes $\widehat{P}(f)$ for $f \in F_B(m)$. This last fact follows from an argument similar to the one given in Sect. 3.2.1.

Maximizing Expected Reward We have shown above that the probability of detection by time T and the negative of mean time to completion are FAB payoff functions. We now show that maximizing expected reward is a FAB payoff function under certain conditions.

Suppose that we obtain reward $r(j,t)$ if we detect the target at time t and in state j for $t = 0, \ldots, T$ and reward $\widehat{r}(T + 1)$ if we do not detect the target by time T. The expected reward from searching with plan f is

$$
\begin{aligned}
&\widehat{R}(f) \\
&= E\left[\sum_{t=0}^{T} r(\omega_t, t)\left(b\left(\zeta\left(\omega, t, f\right) - b\left(\zeta\left(\omega, t-1, f\right)\right)\right) + \widehat{r}(T+1)\left(1 - b\left(\zeta\left(\omega, T, f\right)\right)\right)\right] \\
&= E\left[\sum_{t=0}^{T} \left(r(\omega_t, t) - r(\omega_{t+1}, t+1)\right) b\left(\zeta\left(\omega, t, f\right)\right)\right] + \widehat{r}(T+1)
\end{aligned}
\tag{3.72}
$$

where we define $b\left(\zeta\left(\omega, -1, f\right)\right) = 0$ and $r(\omega_{T+1}, t+1) = \widehat{r}(T+1)$. If we set

$$a_t(\omega) = r(\omega_t, t) - r(\omega_{t+1}, t+1) \quad \text{for} \quad t = 0, \ldots, T,$$

and

$$a_{T+1}(\omega) = \widehat{r}(T+1) \quad \text{for} \quad \omega \in \Omega,$$

then

$$\widehat{R}(f) = E\left[\sum\nolimits_{t=0}^{T} a_t(\omega)\, b\left(\zeta\left(\omega, t, f\right)\right)\right] + \widehat{r}(T+1)$$

is a FAB payoff function.

If $b(z) = 1 - e^{-z}$ and a_t is non-negative for $t = 0, \ldots, T$, then $\widehat{R}(f)$ is concave and we can use the FAB algorithm to find a plan f^* to maximize $\widehat{R}(f)$ over $F_B(m)$. If in addition, the target motion is Markovian, then we may use the Markovian version of the FAB algorithm. Note that the coefficients a_t have distributions that are independent of $X(s)$ for $s < t$ given $X(t)$. To employ this version of the algorithm we calculate

$$E_{jt}[a_t(\omega)] = r(j, t) + \sum\nolimits_{i=1}^{J} \Gamma_t(j, i)\, r(i, t+1) \quad \text{for} \quad j = 1, \ldots, J, \; t = 0, \ldots, T.$$

If we take $r(\omega, t) = t$ for $t = 0, \ldots, T+1$, then $a_t(\omega) = -1$ for $t = 0, \ldots, T$, $a_{T+1}(\omega) = T + 1$ and $\widehat{R}(f)$ is the negative of the mean time to complete the search with plan f.

3.5 Summary

This chapter defines the optimal search for a moving target problem for discrete space and time target motion models. Section 3.1 defines the continuous-effort version of this problem. In the case of a continuous-effort, decreasing-rate detection function, Sect. 3.1.1 finds necessary and sufficient conditions for a plan to be T-optimal within the class $F_B(m)$ of search plans which allocate $m(t)$ effort for

$t = 0, \ldots, T$. It also finds a bound on the probability of detection for the T-optimal plan that can be used to decide when an algorithm has found a plan that is "close enough" to optimal.

Section 3.2 finds algorithms for computing T-optimal plans for a continuous-effort, exponential detection function. This section shows that T-optimal plans may be found by solving a sequence of stationary target problems of the type solved in Chap. 2. The algorithms are recursive and converge to the optimal plan as the number of iterations approach infinity. The bound found in Sect. 3.1.1 is easily computed as part of the recursion and is used to provide a stopping rule for the algorithms. The first pass of the recursion through the times $t = 0, \ldots, T$ produces the myopic plan. This plan allocates effort at time t to maximize the increase in detection probability given failure to detect before time t. Generally the myopic plan is not T-optimal, but it is often close, which means that the recursion tends to converge rapidly to an almost T-optimal plan. A special form of the recursion is given for Markovian target motion. At the end of the section, extensions of the optimal detection search are discussed, including optimal survivor search, optimal defensive search, and optimal whereabouts search.

Section 3.3 finds necessary conditions for the discrete effort search problem. Section 3.4 introduces the FAB algorithm. This algorithm is an extension of the optimal detection search algorithms given in Sect. 3.2 to more general payoff functions such as minimizing mean time to complete a search or maximizing the expected reward from a search. Two versions of the FAB algorithm are presented, one for general target motion and one for Markovian target motion.

3.6 Notes

In 1975 when the first edition of Stone (2007) was published, there were limited results on optimal search for a moving target. Pollock (1970); Dobbie (1974), and Iida (1972) solved two cell problems. Washburn (1975) used a version of the FAB algorithm to solve a 67 cell approximation to the problem of detecting of a target moving according to a diffusion process. The solution converged to a necessary condition for optimality. It was not known at that time that the condition was also sufficient. There were a number of results that obtained necessary conditions. See Lions (1971), Hellman (1972), and Saretsalo (1973). In addition, one could obtain explicit solutions for a class of target motion models called conditionally deterministic. See Stone and Richardson (1974).

Brown (1977), (1980) found necessary and sufficient conditions for a T-optimal search plan for a continuous-effort exponential detection function and an arbitrary discrete space and time target motion process. The conditions state that a plan $f^* \in F(m)$ is T-optimal if and only if for each $t = 0, \ldots, T, f(\cdot, t)$ is the optimal allocation of $m(t)$ effort for the stationary target problem of detecting a target whose distribution is given by the posterior distribution on the target's location at time t given failure to detect by the plan f^* at all times other than t. Brown outlined an iterative algorithm that converges to the T-optimal plan as the number of iterations goes to infinity. In the case of a Markovian target model, he presented an efficient implementation of the recursive algorithm. The FAB algorithm in Washburn (1983) is a generalization of Brown's algorithm for Markov target motion. Washburn (1981) found the upper bound given in Sect. 3.1.1 that is used in the stopping rule for the iterative algorithms given in this chapter.

References

Bertsekas DP (1999) Nonlinear programming, 2nd edn. Athena Scientific, Belmont

Brown SS (1977) Optimal and near optimal searches for a target with multiple scenario Markovian, constrained Markovian, or geometric memory motion in discrete space and time. Daniel H. Wagner, Associates Memorandum Report

Brown SS (1980) Optimal search for a moving target in discrete time and space. Oper Res 28:1275–1289

Crisan D (2001) Particle filters – a theoretical perspective. In: Doucet A, de Freitas N, Gordon N (eds) Sequential Monte Carlo methods in practice. Springer, New York, pp 17–41

Dobbie J (1974) A two-cell model of search for a moving target. Oper Res 22:79–92

Hellman O (1972) On the search for a randomly moving target. SIAM J Appl Math 22:545–552

Iida K (1972) The optimal distribution of searching effort for a moving target. Keiei Kagaku (Japanese) 16:204–215

Lions JL (1971) Optimal control of systems governed by partial differential equations. Springer, New York, pp 248–249

Pollock SM (1970) A simple model of search for a moving target. Oper Res 18:883–903

Saretsalo L (1973) On the optimal search for a target whose motion is a Markov process. J Appl Probability 10:747–756

Shapiro A, Dentcheva D, Ruszczynski L (2009) Lectures on stochastic programming, modeling, and theory. SIAM, Philadelphia

Stone LD (2007) Theory of optimal search, 2nd edn. INFORMS, Catonsville

Stone LD, Richardson HR (1974) Search for targets with conditionally deterministic motion. SIAM J Appl Math 27:239–255

Washburn (1975) Search for a moving diffusing target. ORSA/TIMS conference, Las Vegas

Washburn AR (1980) On search for a moving target. Naval Res 27:315–322

Washburn AR (1981) An upper bound useful in optimizing search for a moving target. Oper Res 29:1227–1230

Washburn AR (1983) Search for a moving target: the FAB algorithm. Oper Res 31:739–751

Washburn AR (2014b) Search and detection, 5th edn. Create Space, North Charleston

Chapter 4
Path-Constrained Search in Discrete Time and Space

In some practical situations, a searcher might have difficulties with implementing an optimal search plan of the form stipulated in the previous chapters. The plan might call for an instantaneous shift of search effort from one time period to the next. If the searcher requires a significant amount of time to carry out this shift, a relatively fast moving target would “get ahead” of the searcher. This situation is especially prevalent in robotic searches of buildings, where transit from room to room accounts for the majority of time expenditure, and searches using low-speed unmanned aerial systems, where the ratio of searcher speed to target speed is low. In this chapter, we describe methods for computing optimal search plans while accounting for real-world constraints on the agility of the searcher. In fact, we consider multiple searchers, each providing a discrete search effort, as well as multiple targets. The chapter starts, however, with the simpler situation of a single searcher looking for a single target. We formulate the optimal search problem as that of finding the optimal searcher path and describe a branch-and-bound algorithm for its solution. We proceed by generalizing the formulation to account for a searcher that operates at different “altitudes” with a more complex sensor. We also describe algorithmic enhancements that both handle the more general situation and provide computational speed-ups. The chapter then addresses the situation with multiple searchers, first of identical types and second of different types and also with multiple targets. These generalizations are most easily handled within a mathematical programming framework, which facilitates the consideration of a multitude of constraints including those related to airspace deconflication and also allows the leverage of well-developed optimization solvers for the determination of optimal searcher plans. The chapter ends with a description of some algorithms behind these solvers, with an emphasis on cutting-plane methods. Throughout the chapter we remain in the context of discrete time and space search.

L.D. Stone et al., *Optimal Search for Moving Targets*, International Series
in Operations Research & Management Science 237,
DOI 10.1007/978-3-319-26899-6_4

4.1 Path-Based Formulation and Solution for Single Searcher

This section provides an introduction to the subject of path-constrained search. We consider the simplest possible situation: a single searcher moves between a discrete set of cells. In each discrete time period, the searcher examines the cell it occupies with the hope of finding a target that similarly moves between the cells. The searcher cannot move freely between cells, but must in each time period select a next cell to move to from a subset of cells. Since the subset might be the cells physically adjacent to the current cell, the sequence of cells visited by the searcher can be thought of as a path in space. We start with the formulation of the problem of finding the path that maximizes the probability of detecting the target and then proceed with algorithms and enhancements.

4.1.1 Path-Centered Formulation

As in previous chapters, we let the search take place in a finite set of cells $\mathscr{J} = \{1,\ldots,J\}$ and over a finite set of time periods $\mathscr{T} = \{1,2,\ldots,T\}$. We let $t = 0$ represent the time prior to search and set $\mathscr{T}_0 = \{0\} \cup \mathscr{T}$. There is one target occupying one cell in each time period. We assume that the target moves according to a Markov chain with transition probability matrix Γ with elements $\Gamma(j,j')$, $j,j' \in \mathscr{J}$. Specifically, $\Gamma(j,j')$ is the probability that a target occupying cell j in time period t occupies cell j' in time period $t+1$. We refer to this as a Markovian target model. Although, this model is quite flexible and offers significant computational advantages as will be clear below, it assumes independence between moves. In Sect. 4.2, we examine a more general (and also computationally more costly) target movement model used by the U.S. Coast Guard's decision aid SAROPS; see Kratzke et al. (2010).

We consider one searcher that during each time $t \in \mathscr{T}_0$ occupies a cell. When in a cell j, the searcher can only move to any cell "adjacent" to j as defined by the forward star $\mathscr{F}(j) \subset \mathscr{J}$. By convention, $j \in \mathscr{F}(j)$ and the searcher can therefore remain in its current cell. We assume there is no transit time between adjacent cells. The situation with nonzero transit time between cells can be modeled, at least approximately, by introducing artificial cells, and also by the more general formulations in Sect. 4.2. We denote the searcher's cell prior to time period 1 by $j_0 \in \mathscr{J}$.

The searcher is equipped with one imperfect sensor. Each time period $t \in \mathscr{T}$, the searcher's sensor takes one "look" in the cell. The probability that one look for the target in cell j during time period t detects the target, given it is actually in the cell, is $g(j,t) \in [0,1)$. We refer to this probability as the *glimpse-detection probability*. The sensor returns no false positives.

For any $t \in \mathscr{T}$ and $j_l \in \mathscr{J}$, $l = 0, 1, 2, \ldots, t$, with $j_l \in \mathscr{F}(j_{l-1})$ for all $l = 1, 2, \ldots, t$, let the sequence $\{j_l\}_{l=0}^{t}$ denote a directed j_0-j_t subpath. If $t = T$, then the directed j_0-j_t subpath is a directed j_0-j_t path that extends across the time horizon. When no misunderstanding can arise, we refer to a directed j_0-j_t (sub)path as a (sub)path. Since $j_l \in \mathscr{F}(j_{l-1})$, the (sub)path is a sequence of adjacent cells that is considered feasible for the searcher to visit and is therefore a candidate search plan.

In this notation, the searcher moves from j_0 to some j_T along a directed j_0-j_T path. The searcher occupies only one cell $j \in \mathscr{J}$ each time period, and stays at the same cell or moves to another cell in $\mathscr{F}(j)$ for the next time period. Consequently, the searcher is *path constrained*.

For any $t \in \mathscr{T}$, let

$$q(\cdot, t) = [q(1, t), q(2, t), \ldots, q(J, t)], \tag{4.1}$$

where $q(j, t)$ is the probability that the target occupies cell $j \in \mathscr{J}$ during time period $t \in \mathscr{T}$ and the target is not detected before t. The initial target distribution $q(\cdot, 1)$ is assumed known. For example, $q(j, 1) = 1$ and $q(j', 1) = 0$ for $j' \neq j$, which implies that the target is certainly in cell j at time 1. Even in this case the search plan could be nontrivial as there might be no guarantee that the searcher detects the target in the first time period and after that the movement of the target might be uncertain.

We refer to $q(\cdot, t)$ as the undetected target distribution, but note that $q(\cdot, t)$ is not a probability mass function and might be considered a "defective probability distribution." We stress that $q(j, t)$ differs from the probability that the target occupies cell $j \in \mathscr{J}$ during time period $t \in \mathscr{T}$ *given* that the target is not detected before t. The latter probability represents the Bayesian posterior probability of target location at the beginning of time period t. However, as seen in the next paragraphs, it is not necessary to consider that probability.

We express the probability of detection directly in terms of $q(\cdot, \cdot)$. In this notation, the probability of detection in cell j during time period t and no prior detections becomes

$$q(j, t) g(j, t). \tag{4.2}$$

Since $q(\cdot, t)$ is the undetected target distribution at the beginning of time period t, it depends on searches up to time period $t - 1$. Specifically, if cell j is searched during time period t, then

$$q(\cdot, t + 1) = [q(1, t), .., q(j - 1, t), q(j, t)(1 - g(j, t)), q(j + 1, t), .., q(J, t)]\Gamma. \tag{4.3}$$

Given a path $\mathscr{P} = \{j_t\}_{t=0}^{T}$, the T events "detection during time period t and target is not detected before t," $t = 1, 2, \ldots, T$, are mutually exclusive. Hence, in view of (4.2), the probability of detection along $\mathscr{P}$, denoted $P(\mathscr{P})$, becomes

$$P(\mathscr{P}) = \sum_{t=1}^{T} q(j_t, t) g(j_t, t). \tag{4.4}$$

The challenge is then how to efficiently compute $P(\mathscr{P})$ for a large number of (relevant) paths and to ensure that the paths not examined are inferior. This is the subject dealt with next, which then leads to an algorithm for finding a path that maximizes this probability.

Example of Path-Constrained Search

We consider a simple example with an area of interest divided into four cells numbered 1–4 from the top-left to the bottom-right. It is known that the target in time period 1 is in cell 2 with probability 0.6 and in cell 3 with probability 0.4. Consequently,

$$q(\cdot, 1) = [0 \;\; 0.6 \;\; 0.4 \;\; 0].$$

The target moves vertically with probability 0.8 and horizontally with probability 0.2. The target cannot remain in the same cell for two consecutive time periods. This gives a Markov transition matrix

$$\Gamma = \begin{bmatrix} 0 & 0.2 & 0.8 & 0 \\ 0.2 & 0 & 0 & 0.8 \\ 0.8 & 0 & 0 & 0.2 \\ 0 & 0.8 & 0.2 & 0 \end{bmatrix}.$$

Consequently,

$$q(\cdot, 1)\Gamma = [0.440 \;\; 0 \;\; 0 \;\; 0.560],$$

which gives the probability distribution of the target during time period 2 in the absent of information collected from search.

Now suppose that we have one searcher occupying cell 1 at time period 0. It can move vertically or horizontally, but not diagonally. Thus,

$$\mathscr{F}(1) = \{1, 2, 3\}, \mathscr{F}(2) = \{1, 2, 4\}, \mathscr{F}(3) = \{1, 3, 4\}, \mathscr{F}(4) = \{2, 3, 4\}.$$

We let the time horizon $T = 2$. Then, the searcher has the following possible paths for time periods 0,1,2:

$$\{1, 1, 1\}, \{1, 1, 2\}, \{1, 1, 3\}, \{1, 2, 1\}, \{1, 2, 2\}, \{1, 2, 4\}, \{1, 3, 1\}, \{1, 3, 3\}, \{1, 3, 4\},$$

Let the glimpse-detection probability $g(j, t) = 0.9$ for all j and t. The probability $q(\cdot, 2)$ (see (4.3)) depends on the cell in which search took place during time period 1. We consider all three possibilities:

search cell 1 at $t = 1$: $q(\cdot, 2) = [0 \cdot (1 - 0.9) \;\; 0.6 \;\; 0.4 \;\; 0]\Gamma = [0.440 \;\; 0 \;\; 0 \;\; 0.560]$,

search cell 2 at $t = 1$: $q(\cdot, 2) = [0 \;\; 0.6 \cdot (1 - 0.9) \;\; 0.4 \;\; 0]\Gamma = [0.332 \;\; 0 \;\; 0 \;\; 0.128]$,

search cell 3 at $t = 1$: $q(\cdot, 2) = [0 \;\; 0.6 \;\; 0.4 \cdot (1 - 0.9) \;\; 0]\Gamma = [0.152 \;\; 0 \;\; 0 \;\; 0.488]$.

The probability of detection along the nine paths are therefore (see (4.4)):

$$\begin{aligned}
&\text{path } \{1,1,1\}: \; 0 \cdot 0.9 + 0.44 \cdot 0.9 = 0.396, \\
&\text{path } \{1,1,2\}: \; 0 \cdot 0.9 + 0 \cdot 0.9 = 0, \\
&\text{path } \{1,1,3\}: \; 0 \cdot 0.9 + 0 \cdot 0.9 = 0, \\
&\text{path } \{1,2,1\}: \; 0.6 \cdot 0.9 + 0.332 \cdot 0.9 = 0.839, \\
&\text{path } \{1,2,2\}: \; 0.6 \cdot 0.9 + 0 \cdot 0.9 = 0.540, \\
&\text{path } \{1,2,4\}: \; 0.6 \cdot 0.9 + 0.128 \cdot 0.9 = 0.655, \\
&\text{path } \{1,3,1\}: \; 0.4 \cdot 0.9 + 0.152 \cdot 0.9 = 0.497, \\
&\text{path } \{1,3,3\}: \; 0.4 \cdot 0.9 + 0 \cdot 0.9 = 0.360, \\
&\text{path } \{1,3,4\}: \; 0.4 \cdot 0.9 + 0.488 \cdot 0.9 = 0.799.
\end{aligned}$$

We see that path $\{1,2,1\}$ is optimal with probability of detection of 0.839. It is interesting to note that this search plan first attempts to detect the target in cell 2, which is the most likely location for the target at that time. Second, the search plan prescribes cell 1, which is not the most likely location of the target at $t = 2$. The target is in cell 1 with probability 0.44 and cell 4 with probability 0.560. However, given that the searcher did not detect the target during time period 1 in cell 2, we view it less likely that the target was even in cell 2 at that time. It becomes more likely that the target actually was in cell 3 during time period 1. Specifically, the probability that the target is in cell 1 during time period 2 and the target was not detected during time period 1 is $q(1,2) = 0.332$. For cell 4 we have similarly that $q(4,2) = 0.128$. In view of this calculation, it is clear that moving back to cell 1 for time period 2 is better than moving to cell 4. In fact, it is optimal.

4.1.2 Branch-and-Bound Algorithm

It is clear that an algorithm for maximizing $P(\mathscr{P})$ can be based on enumerating paths and checking the associated probability of detection. However, the number of possible paths is typically astronomical. Even in the limited case of $\mathscr{F}(j)$ consisting of only five cells (maybe cell j as well as the four cells immediately north, south, east, and west of j in a regular grid) and a time horizon of $T = 12$, there are more than 244 million possible paths. A branch-and-bound algorithm enumerates a subset of these paths and uses bounds to conclude that those not examined must have no higher probability of detection; see Rardin (1997) for an elementary introduction to branch-and-bound algorithms. The simplest version of such an algorithm is described next. Section 4.1.4 provides several enhancements.

4.1.2.1 Basic Algorithm

Given a subpath $\{j_l\}_{l=0}^{t}$, $t \in \mathscr{T}$, we let $\bar{p}(j_t, t)$ denote an upper bound on the probability of detection along any path that starts with the subpath $\{j_l\}_{l=0}^{t}$. We note that to simplify the notation we specify only the last cell j_t in the notation $\bar{p}(j_t, t)$ even though the quantity could depend on the whole subpath $\{j_l\}_{l=0}^{t}$. Possible ways of computing such an upper bound are given below and indeed will become the central challenge in a branch-and-bound algorithm. We define

$$\mathscr{K}(t) \text{ to be the set of triplets of the form } (j_t, t, \bar{p}(j_t, t)), \tag{4.5}$$

representing one-cell extensions of $\{j_l\}_{l=0}^{t-1}$ yet to be explored. The first element j_t refers to the next cell to visit and the second element t is the time period[1] to visit cell j_t.

The upper bound $\bar{p}(j_t, t)$ is the sum of three parts. Let $d_t(j_t, t)$ be an upper bound on the probability of detection during time periods $t+1, t+2, \ldots, T$ and no detection along subpath $\{j_l\}_{l=0}^{t}$ given that the searcher is at j_t during time period t. The two other parts are the probability of detection along the subpath $\{j_l\}_{l=0}^{t-1}$ and the probability of detection during t. Hence,

$$\bar{p}(j_t, t) = P(\{j_l\}_{l=0}^{t-1}) + q(j_t, t)g(j_t, t) + d_t(j_t, t). \tag{4.6}$$

We also let $\hat{p}$ denote the largest detection probability of all the examined paths at a given stage of the algorithm. In this notation, a branch-and-bound algorithm takes the following form where details about bound calculations are presented below.

Basic Branch-and-Bound Algorithm.

Step 0. Set $t = 0$, $\mathscr{K}(0) = \{(j_0, 0, 1)\}$, and $\hat{p} = 0$.
Step 1. If $\mathscr{K}(t)$ is empty, replace t by $t-1$. Else, go to Step 3.
Step 2. If $t = 0$, stop: the last saved path is optimal and $\hat{p}$ is its probability of detection. Else, go to Step 1.
Step 3. Remove from $\mathscr{K}(t)$ the triplet $(j_t, t, \bar{p}(j_t, t))$ with the largest bound $\bar{p}(j_t, t)$.
Step 4. If $\bar{p}(j_t, t) \leq \hat{p}$, go to Step 1. (Current subpath is fathomed.)
Step 5. If $t < T$, then for each cell $j \in \mathscr{F}(j_t)$, calculate a bound $d_{t+1}(j, t+1)$ as well as $\bar{p}(j, t+1)$, see (4.6), and add $(j, t+1, \bar{p}(j, t+1))$ to $\mathscr{K}(t+1)$. Replace t by $t+1$ and go to Step 3. Else, let $\hat{p} = \bar{p}(j_t, t)$ and save the incumbent path $\{j_l\}_{l=0}^{T}$, and go to Step 1.

[1]This information is currently redundant but the notation is convenient in later generalizations.

The algorithm starts with a single element in $\mathscr{K}(0)$, which is immediately removed in Step 3. Step 4 is at first not invoked and the algorithm moves to Step 5. Here, a bound is computed for every cell to which the searcher might move from its initial cell j_0 and the set $\mathscr{K}(1)$ is populated. The time is advanced to $t = 1$ and the algorithm moves to Step 3. There the most promising extension is examined first, i.e., the one with the largest bound. Presumably, a large bound indicates that the probabilities of detection along the corresponding paths are high. Of course, there is no guarantee that this is actually the case, but the rule is reasonable. Step 4 is again skipped and we move to Step 5. This process is repeated until the algorithm has constructed a path across the time horizon and $t = T$ in Step 5. This path then becomes the incumbent path whose probability of detection is the best seen thus far. The algorithm then moves to Step 1, where either additional elements of $\mathscr{K}(t)$ are examined or time is reduced. In the absence of Step 4, this process would have led to a complete enumeration of all paths. However, Step 4 inserts a check that prevents an extension of a subpath whose bound is no greater than the probability of detection for the incumbent path. Since the bound provides an optimistic view of what can be achieved by extending the subpath, every extension of such a subpath must result in a probability of detection that is no greater than that of the incumbent. Any extension of such a subpath can therefore be ignored. This "fathoming" of a subpath may reduce the amount of path enumeration dramatically. In any case, the algorithm terminates with an optimal path.

Clearly, a tight bound $d_t(j_t, t)$ increases the number of times the algorithm fathoms and reduces the computing time. We here only describe the simplest idea for computing such a bound and defer to Sect. 4.1.4 for enhancements.

4.1.2.2 Mean Bound Calculation

The only challenge with computing the bound in (4.6) is associated with the last term on the right-hand side. The two first terms on that side are, of course, known at the relevant step of the algorithm. We rely on the relationship between probabilities and expectations as described next.

An upper bound $d_t(j_t, t)$ on the probability of detection during time periods $t + 1, t + 2, \ldots, T$ and no detection along subpath $\{j_l\}_{l=0}^{t}$ given that the searcher is at j_t during time period t is furnished by the largest possible *expected number of detections* during time periods $t + 1, t + 2, \ldots, T$. This fact is most easily realized by observing that the expected number of detections is simply the sum of the probabilities of detection in each time period. If the path is known, each term in this sum is simply a product of glimpse-detection probability and the probability that the target is in the corresponding cell. The latter is given by the Markov transition matrix Γ and the initial probability distribution $q(\cdot, 1)$.

Two complications arise, however. First, the bound $d_t(j_t, t)$ should account for the fact that no detection along subpath $\{j_l\}_{l=0}^{t}$ has taken place. This provides, indirectly, additional information about the location of the target. Second, there are typically many ways of extending the subpath to a path, and we need to consider the one with the largest expected number of detections. We overcome these challenges in the manner described next.

We let $q_g(j,t)$ be the probability that the target occupies cell j during time period t and not detected along the subpath $\{j_l\}_{l=0}^t$, $t \in \mathscr{T}$, i.e.,

$$q_g(\cdot,t) = [q(1,t),\ldots,q(j_t-1,t),q(j_t,t)(1-g(j_t,t)),q(j_t+1,t),\ldots,q(J,t)]. \tag{4.7}$$

We use subscript g to indicate that $q_g(\cdot,t)$ is obtained from $q(\cdot,t)$ by applying the glimpse-detection probability corresponding to the last cell in $\{j_l\}_{l=0}^t$. For any integer $s > t$, $s,t \in \mathscr{T}$, we also define

$$q_\Gamma(\cdot,s;t) = q_g(\cdot,t)\Gamma^{s-t}. \tag{4.8}$$

As seen, $q_\Gamma(j,s;t)$ is the probability that the target occupies cell j during time period s and no detection along subpath $\{j_l\}_{l=0}^t$. In contrast to $q(\cdot,s)$, $q_\Gamma(\cdot,s;t)$ ignores the effect of search after time period t. The probabilities $q_\Gamma(j,s;t)$, $j \in \mathscr{J}$, provides a revised estimate of where the target is at time s, utilizing the (nondetection) information collected up to time t. This takes care of the first issue described above. If the subpath is simply $\{j_0\}$, i.e., $t = 0$, we define for notational convenience $q_\Gamma(\cdot,s;0) = q(\cdot,1)\Gamma^{s-1}$, for any $s \in \mathscr{T}$, and $q_\Gamma(j,t;t) = 0$ for all $j \in \mathscr{J}$ and $t \in \mathscr{T}_0$. Consequently, $q_\Gamma(\cdot,s;0)$ specifies the probabilities for the target location in the *absence* of information from search. These probabilities represent our knowledge about time period s prior to any search. In contrast, $q_\Gamma(\cdot,s;t)$ represents the updated information after searching the subpath $\{j_l\}_{l=0}^t$ unsuccessfully.

To deal with the second challenge of how to determine the extension of the subpath $\{j_l\}_{l=0}^t$ to a path that has the largest expected number of detections we construct a time-expanded network. Each cell $j \in \mathscr{J}$ is duplicated T times to define the nodes $\langle j,t\rangle$, $t \in \mathscr{T}$. Let $\mathscr{N}$ be the set of all such nodes as well as the nodes $n_0 = \langle j_0,0\rangle$ and $\hat{n} = \langle \hat{j}, T+1\rangle$ representing the searcher's prior position and final position, respectively. Here, $\hat{j}$ is an artificial terminal cell. Two nodes $n = \langle j,t-1\rangle$ and $n' = \langle j',t\rangle$, $j,j' \in \mathscr{J}$ and $t = 2,3,\ldots,T$, are connected with an arc (n,n') if and only if $j' \in \mathscr{F}(j)$. Moreover, the node $n_0 = \langle j_0,0\rangle$ is connected with an arc to a node $n' = \langle j',1\rangle$, $j' \in \mathscr{J}$, if and only if $j' \in \mathscr{F}(j_0)$; and every node $n = \langle j,T\rangle$, $j \in \mathscr{J}$ is connected with an arc to $\hat{n}$. Let $\mathscr{A}$ be the set of all arcs. For any integer $t \le T+1$ and nodes $n_l = \langle j_l,l\rangle \in \mathscr{N}$, $l = 0,1,\ldots,t$, such that $(n_{l-1},n_l) \in \mathscr{A}$ for all $l = 1,2,\ldots,t$, we let the sequence $\{n_l\}_{l=0}^t$ denote a subpath in the time-expanded graph $(\mathscr{N},\mathscr{A})$.

For some $t \in \{0,1,\ldots,T-1\}$, suppose that a subpath $\{j_l\}_{l=0}^t$ is given. Then, we endow each arc $(n,n') = (\langle j,s\rangle,\langle j',s+1\rangle) \in \mathscr{A}$, $s = t,t+1,\ldots,T-1$, in the time-expanded graph $(\mathscr{N},\mathscr{A})$ with a "reward"

$$c_{n,n'} = q_\Gamma(j',s+1;t)g(j',s+1), \tag{4.9}$$

where $\Gamma(j, j')$ is the j-j' element of the Markov transition matrix Γ. We set $c_{n,\hat{n}} = 0$ for all $(n, \hat{n}) \in \mathscr{A}$. In view of (4.8), we see that $c_{n,n'}$ is the probability of detection during time period $s+1$ *and* no detection along subpath $\{j_l\}_{l=0}^t$. We refer to $(\mathscr{N}, \mathscr{A})$ with arc rewards given by (4.9) as the time-expanded network.

We can show that given the subpath $\{j_l\}_{l=0}^t$ the optimal value of the longest-path problem in the time-expanded network from node $\langle j_t, t\rangle$ to node $\langle \hat{j}, T+1\rangle$, using the rewards in (4.9) as "arc length," provides an upper bound on the probability of detection during time periods $t+1, t+2, \ldots, T$ and no detections along the subpath $\{j_l\}_{l=0}^t$ given that the searcher is at j_t during time period t. It represents the expected number of detections during time periods $t+1, t+2, \ldots, T$ given no detections along the subpath $\{j_l\}_{l=0}^t$ and that the searcher is at j_t during time period t. We denote this bound by $d_t(j_t, t)$ and refer to it as the dynamic bound as it needs to be recomputed every time the *current* subpath is extended in the basic algorithm. The bound is called the mean bound.

Since the time-expanded network is acyclic, the longest-path problem can be solved in polynomial calculation time with a standard shortest-path algorithm (Ahuja et al. 1993, pp. 77–79). We observe that to compute $d_t(j_t, t)$ given the subpath $\{j_l\}_{l=0}^t$, it is only necessary to generate the part of the graph $(\mathscr{N}, \mathscr{A})$, and the corresponding arc rewards, "after" time t and within reach from node $\langle j_t, t\rangle$, since the longest path starts at node $\langle j_t, t\rangle$.

4.1.3 Model Enhancements

There are several possible extensions of the model of the previous subsection. We here describe two possibilities that are practically useful. The first one allows the searcher to operate in a different space than the target. For example, the target might operate in an area on the ground while the searcher moves about in the 3-dimensional airspace above the area. This provides substantial modeling flexibility.

The second enhancement makes the glimpse-detection probability dependent on the previous as well as the current cell the searcher occupies. This dependence may arise if adjusting search pattern and/or altitude, refocusing a sensor, and becoming familiar with a new cell have a significant detrimental effect on the searcher's capability to detect a target. In addition, this dependance allows us to account indirectly for small transit times (much less than the length of a time period) between cells by reducing the glimpse-detection probability from its nominal value if the searcher just moved into a cell. For example, suppose that the real-world travel time from cell j' to j is 1 min. To model this situation (approximately), we would normally require a time period of (approximately) 1-min duration. However, this may result in a large number of time periods and long computing times. Alternatively, we can define a longer time period, say 10 min, and let the glimpse-detection probability in cell j be somewhat reduced if a searcher's previous cell were j' as compared to if it were j. This will reflect the fact that a searcher coming from cell j' has only 9 min to search j compared to 10 min if the searcher had already been present in j. Hence, we avoid adopting a fine time discretization with resulting high computational cost.

Each cell $j \in \mathscr{J}$ is associated with a set $\mathscr{H} = \{1, 2, \ldots, H\}$, which can be thought of as various altitudes above the cell. However, the meaning will vary with the application setting. For simplicity we let $\mathscr{H}$ be independent of j, but this is only for notational convenience. For any $j \in \mathscr{J}$ and $h \in \mathscr{H}$, we refer to the cell-altitude pair $\langle j, h \rangle$ as a waypoint where the searcher can "loiter" and carry out search of cell j. We model the area of operations for the searcher by a directed network $(\mathscr{V}, \mathscr{E})$, with set of vertices $\mathscr{V}$ and set of directed edges $\mathscr{E}$, in which vertices $v = \langle j, h \rangle \in \mathscr{V}$ represent waypoints and directed edges $e = (v, v') \in \mathscr{E}$ represent transition between waypoints $v, v' \in \mathscr{V}$. The searcher can only transit between two waypoints that are "adjacent" to each other. Let $\mathscr{F}(v) \subset \mathscr{V}$ be the set of vertices that are adjacent to $v \in \mathscr{V}$. We refer to $\mathscr{F}(v)$ as the forward star of vertex v. We adopt the convention that $v \in \mathscr{F}(v)$ for all $v \in \mathscr{V}$. Then, the set of edges $\mathscr{E} = \{(v, v') \in \mathscr{V} \times \mathscr{V} | \; v' \in \mathscr{F}(v)\}$.

During each time period $t \in \mathscr{T}$, the searcher is at a particular vertex (waypoint). We assume there is no transit time between waypoints. Hence, $(v, v') \in \mathscr{E}$ simply represents search at waypoint v followed by search at waypoint v' in the next time period. As above, the situation with nonzero transit time between waypoints can be modeled, at least approximately, by introducing artificial vertices.

Let $\phi : \mathscr{V} \rightarrow \mathscr{J}$ be the function that specifies the cell with which a vertex is associated, i.e., cell $\phi(v)$ is searched from vertex v. We denote the searcher's vertex prior to time period 1 by $v_0 \in \mathscr{V}$.

For any $t \in \mathscr{T}$ and $v_l \in \mathscr{V}$, $l = 0, 1, 2, \ldots, t$, such that $(v_{l-1}, v_l) \in \mathscr{E}$ for all $l = 1, 2, \ldots, t$, let the sequence $\{v_l\}_{l=0}^{t}$ denote a directed v_0-v_t subpath. If $t = T$, then the directed v_0-v_t subpath is a directed v_0-v_t path that extends across the time horizon. When no misunderstanding can arise, we refer to a directed v_0-v_t (sub)path as a (sub)path. In this notation, the searcher flies from v_0 to some v_T along a directed v_0-v_T path. The searcher occupies only one vertex $v \in \mathscr{V}$ each time period, and stays at the same vertex or moves to another vertex in $\mathscr{F}(v)$ for the next time period.

We let $g(v, v', t)$ be the probability that the searcher at waypoint v' detects the target during time period t, given that the target occupies cell j during t, with $\phi(v') = j$, and that the searcher was at waypoint v during $t - 1$. Again, we call $g(v, v', t)$ a glimpse-detection probability. Since we now allow for the glimpse-detection probability to depend on the vertex and not only the cell, we can capture situations where the sensor performance depends on a factor such as searcher altitude. Moveover, the glimpse-detection probability also depends on the previous vertex with possible benefits described above.

In this notation, the probability of detection at waypoint v' during time period t and no prior detections becomes

$$q(\phi(v'), t) g(v, v', t) \tag{4.10}$$

given search at waypoint v during time period $t - 1$.

Similar to above, if cell j' is searched from waypoint v' during time period t, then

$$q(\cdot, t+1) = [q(1,t), .., q(j'-1,t), q(j',t)(1-g(v,v',t)), q(j'+1,t), .., q(J,t)]\Gamma, \tag{4.11}$$

where v is the searcher's vertex during time period $t-1$.

Given a path $\mathscr{P} = \{v_t\}_{t=0}^{T}$, the T events "detection during time period t and target is not detected before t," $t = 1, 2, \ldots, T$, are mutually exclusive. Hence, in view of (4.10), the probability of detection along $\mathscr{P}$, again denoted $P(\mathscr{P})$, becomes

$$P(\mathscr{P}) = \sum_{t=1}^{T} q(\phi(v_t), t) g(v_{t-1}, v_t, t). \tag{4.12}$$

As in the simpler case above, the challenge then becomes how to efficiently compute $P(\mathscr{P})$ for a large number of (relevant) paths and to ensure that the paths not examined must be inferior.

4.1.4 *Algorithmic Improvements*

The above model enhancements certainly require algorithmic adjustments. Moreover, we also would like to consider algorithmic improvements that could speed up calculations regardless of these modeling changes. We next describe some of the possibilities. The remained of this section can be skip by a reader that is less interested in algorithmic details.

4.1.4.1 Algorithmic Framework

We start by giving an extension of the basic algorithm that applies to the enhanced model. In essence, we only need to replace cells by vertices. Given a subpath $\{v_l\}_{l=0}^{t}$, $t \in \mathscr{T}$, we let $\bar{p}(v_t, t)$ denote an upper bound on the probability of detection along any path that starts with the subpath $\{v_l\}_{l=0}^{t}$. Similar to before we define $\mathscr{K}(t)$ to be the set of triplets of the form $(v_t, t, \bar{p}(v_t, t))$ representing extensions of $\{v_l\}_{l=0}^{t-1}$ yet to be explored. The first element v_t refers to the next vertex to visit and the second element t is the time period to visit vertex v_t. The upper bound $\bar{p}(v_t, t)$ consists of three parts. Let $d_t(v_t, t)$ be an upper bound on the probability of detection during time periods $t+1, t+2, \ldots, T$ and no detection along subpath $\{v_l\}_{l=0}^{t}$ given that the searcher is at v_t during time period t. The two other parts are the probability of detection along the subpath $\{v_l\}_{l=0}^{t-1}$ and the probability of detection during t. Hence,

$$\bar{p}(v_t, t) = P(\{v_l\}_{l=0}^{t-1}) + q(\phi(v_t), t) g(v_t, t) + d_t(v_t, t). \tag{4.13}$$

In this notation, the enhanced branch-and-bound algorithm takes the following form.

Enhanced Branch-and-Bound Algorithm.

Step 0. Set $t = 0$, $\mathscr{K}(t) = \{(v_0, 0, 1)\}$, and $\hat{p} = 0$.
Step 1. If $\mathscr{K}(t)$ is empty, replace t by $t - 1$. Else, go to Step 3.
Step 2. If $t = 0$, stop: the last saved path is optimal and $\hat{p}$ is its probability of detection. Else, go to Step 1.
Step 3. Remove from $\mathscr{K}(t)$ the triplet $(v_t, t, \bar{p}(v_t, t))$ with the largest $\bar{p}(v_t, t)$.
Step 4. If $\bar{p}(v_t, t) \leq \hat{p}$, go to Step 1. (Current subpath is fathomed.)
Step 5. If $t < T$, then for each $v \in \mathscr{F}(v_t)$, calculate $d_{t+1}(v, t + 1)$ and $\bar{p}(v, t + 1)$, and add $(v, t + 1, \bar{p}(v, t + 1))$ to $\mathscr{K}(t + 1)$. Replace t by $t + 1$ and go to Step 3. Else, let $\hat{p} = \bar{p}(v_t, t)$ and save the path $\{v_l\}_{l=0}^{T}$, and go to Step 1.

We describe various bounding techniques next.

4.1.4.2 Enhanced Dynamic Bound Calculations

We can easily extend the dynamic bound to the present case. Similar to the above development, we let $q_g(j, t)$ be the probability that the target occupies cell j during time period t and no detected along the subpath $\{v_l\}_{l=0}^{t}$, $t \in \mathscr{T}$, i.e.,

$$q_g(\cdot, t) = \tag{4.14}$$

$$[q(1, t), \ldots, q(\phi(v_t) - 1, t), q(\phi(v_t), t)(1 - g(v_{t-1}, v_t, t)), q(\phi(v_t) + 1, t), \ldots, q(J, t)].$$

For any integer $s > t$, $s, t \in \mathscr{T}$, we let $q_\Gamma(\cdot, s; t)$ be defined as above.

We construct a time-expanded graph from the network $(\mathscr{V}, \mathscr{E})$ as follows. Each vertex $v \in \mathscr{V}$ is duplicated T times to define the nodes $\langle v, t\rangle$, $t \in \mathscr{T}$. Let $\mathscr{N}$ be the set of all such nodes as well as the nodes $n_0 = \langle v_0, 0\rangle$ and $\hat{n} = \langle \hat{v}, T + 1\rangle$ representing the searcher's prior position and final position, respectively. Here, $\hat{v}$ is an artificial terminal vertex. Two nodes $n = \langle v, t - 1\rangle$ and $n' = \langle v', t\rangle$, $v, v' \in \mathscr{V}$ and $t = 2, 3, \ldots, T$, are connected with an arc (n, n') if and only if $(v, v') \in \mathscr{E}$. Moreover, the node $n_0 = \langle v_0, 0\rangle$ is connected with an arc to a node $n' = \langle v', 1\rangle$, $v' \in \mathscr{V}$, if and only if $(v_0, v') \in \mathscr{E}$; and every node $n = \langle v, T\rangle$, $v \in \mathscr{V}$ is connected with an arc to $\hat{n}$. Let $\mathscr{A}$ be the set of all arcs. For any integer $t \leq T + 1$ and nodes $n_l = \langle v_l, l\rangle \in \mathscr{N}$, $l = 0, 1, \ldots, t$, such that $(n_{l-1}, n_l) \in \mathscr{A}$ for all $l = 1, 2, \ldots, t$, we let the sequence $\{n_l\}_{l=0}^{t}$ denote a subpath in the time-expanded graph $(\mathscr{N}, \mathscr{A})$.

For some $t \in \{0, 1, \ldots, T - 1\}$, suppose that a subpath $\{v_l\}_{l=0}^{t}$ in the original graph $(\mathscr{V}, \mathscr{E})$ is given. Then, we endow each arc $(n, n') = (\langle v, s\rangle, \langle v', s + 1\rangle) \in \mathscr{A}$, $s = t, t + 1, \ldots, T - 1$, in the time-expanded graph $(\mathscr{N}, \mathscr{A})$ with a "reward"

$$c_{n,n'} = \left[q_\Gamma(\phi(v'), s+1; t) - q_\Gamma(\phi(v), s; t) \left(\min_{v'' \in \mathscr{R}(v)} g(v'', v, s) \right) \Gamma(v, v') \right] g(v, v', s+1), \tag{4.15}$$

where $\Gamma(v, v')$ is the $\phi(v)$-$\phi(v')$ element of the Markov transition matrix Γ and $\mathscr{R}(v) \subset \mathscr{V}$ is the reverse star of v, i.e., $\mathscr{R}(v) = \{v'' \in \mathscr{V} \mid (v'', v) \in \mathscr{E}\}$. The "min" in the formula ensures that the arc reward $c_{n,n'}$ is independent of the previous vertex v'', which would have ruin the longest-path structure of the bound calculation problem: $c_{n,n'}$ would no longer only depend on the head and tail of the arc (n, n'). Hence, it becomes necessary to use this conservative estimate.

We also set $c_{n,\hat{n}} = 0$ for all $(n, \hat{n}) \in \mathscr{A}$. In view of the above development, we see that $c_{n,n'}$ is the probability of detection during time period $s+1$ *and* no detection along subpath $\{v_l\}_{l=0}^t$ *and* no detection during time period s. We refer to $(\mathscr{N}, \mathscr{A})$ with arc rewards given by (4.15) as the time-expanded network.

With this reward, the bound calculation remains a longest-path problem in an acyclic graph and it can be shown using the same arguments as in Lau et al. (2008) that this enhanced dynamic bound $d_t(v_t, t)$ computed from (4.15) indeed is an upper bound on the probability of detection during time periods $t+1, t+2, \ldots, T$ and no detection during the subpath $\{v_l\}_{l=0}^t$ given that the searcher is at v_t during time period t.

4.1.4.3 Static Bound Calculations

In this subsection, we consider an alternative and in fact simpler bound than the (enhanced) dynamic bound $d_t(v_t, t)$.

The (Enhanced) Branch-and-Bound Algorithm with the dynamic bound requires one longest-path calculation in a time-expanded network for each vertex in the forward star of the current vertex to compute the required bounds $d_t(v_t, t)$ (see Step 5). This is aligned with the traditional approach of branch-and-bound algorithms where a bound is reoptimized before each branching. In the present case, the reoptimization corresponds to the longest-path calculation and requires computing the arc rewards $c_{n,n'}$, see (4.15). As an alternative, one can use a static bound, computable prior to any branching as described next.

The dynamic bound $d_t(v_t, t)$ requires knowledge of the current subpath $\{v_l\}_{l=0}^t$ as (4.15) depends on $q_\Gamma(\cdot, \cdot; t)$. Suppose that we ignore that subpath information and compute the optimal value of the longest-path problem as in the case of $d_t(v_t, t)$, but now with $q_\Gamma(\cdot, \cdot; t)$ in (4.15) replaced by $q_\Gamma(\cdot, \cdot; 0)$. Then, as we prove in Theorem 4.1 below, this value is an upper bound on the probability of detection during time periods $t+1, t+2, \ldots, T$, given that the searcher is at v_t during time period t. Hence, that value is also an upper bound on the probability of detection during time periods $t+1, t+2, \ldots, T$ and no detection along the subpath $\{v_l\}_{l=0}^t$, given that the searcher is at v_t during time period t. We refer to that value as the static bound and denote it by $d_0(v_t, t)$, where the subscript 0 indicates that the trivial subpath $\{v_0\}$ is used in (4.15) with $t = 0$ instead of the subpath $\{v_l\}_{l=0}^t$. We note that $c_{n,n'}$ in (4.15) with subpath $\{v_0\}$ is effectively the probability of detection at vertex v' during time period $s + 1$ and no detection at vertex v during time period s. Since $d_0(v_t, t)$ is independent of the current subpath used to reach the vertex v_t it can be computed in advance for all nodes $\langle v, t\rangle \in \mathscr{N}$, and dynamical computation of bounds is not required. Consequently, the arc rewards (4.15) and bounds are computed only once. We observe that it is not necessary to carry out a longest-path calculation from each node $\langle v, t\rangle \in \mathscr{N}$ to $\langle \hat{v}, T + 1\rangle$ to obtain $d_0(v, t)$. It is more efficient to carry out the longest-path calculations backward from node $\langle \hat{v}, T + 1\rangle$ to all nodes. This calculation simply amounts to applying once a shortest-path algorithm to the time-expanded network with arc lengths equal to the negative rewards.

In Step 5 of the Enhanced Branch-and-Bound Algorithm, we now simply use $d_0(v_t, t)$ instead of $d_t(v_t, t)$. Thus, the modified algorithm does not require any longest-path calculation in Step 5. All bound calculations are done prior to Step 0. Clearly, the modified approach results in a weaker bound and more branching attempts are typically needed. However, the additional branching attempts may be compensated by shorter per-iteration computing times. The empirical evidence in Sato and Royset (2010) indeed points in that direction.

4.1.4.4 Directional Static Bound

We also derive a stronger static bound motivated by the classical approach to handle turn-radius constraints in vehicle routing problems as pioneered by Caldwell (1961).

In the longest-path calculations for the static bound, the reward of arc $(\langle v, s\rangle, \langle v', s + 1\rangle)$ is, effectively, the probability of detection at vertex v' during time period $s + 1$ and no detection at vertex v during time period s. We strengthen the static bound if we redefine the arc reward to be, effectively, the probability of detection at vertex v' during time period $s + 1$ and no detection at vertex v during time period s *and* no detection at the vertex visited during time period $s - 1$. However, redefining the arc reward to depend not only on the arc's head and tail nodes, but also on a previous node ruins the longest-path structure of the bound-calculation problem.

A similar situation arises in vehicle routing problems for vehicles with turn-radius constraints or penalties. The classical approach to handle that situation is to duplicate each node a number of times equal to the number of nodes in the node's reverse star. An arc in the resulting "node-expanded" network then carries information about three nodes, not only two, and a desirable network structure of the problem can be maintained. Fortunately, it is often practical to carry out such a node-expansion approach in the present context because the number of nodes in the reverse star is typically quite moderate. Hence, we proceed along the stated lines and develop a node-and-time expanded network, in which the improved static bound can be calculated by solving a longest-path problem. We refer to this improved bound as the directional static bound.

For any $n' \in \mathscr{N}$, let $\mathscr{R}(n') \subset \mathscr{N}$ be the reverse star of n', i.e., $\mathscr{R}(n') = \{n \in \mathscr{N} | (n, n') \in \mathscr{A}\}$. Then, for any $n, n' \in \mathscr{N} \backslash \{\hat{n}\}$ such that $(n, n') \in \mathscr{A}$, we define an expanded node $\xi = \langle n, n' \rangle$. We do not expand the end node, so we set $\hat{\xi} = \hat{n}$. Let Ξ be the set of all expanded nodes. Two expanded nodes $\xi, \xi' \in \Xi$ are connected by an expanded arc (ξ, ξ') if $\xi = \langle n, n' \rangle$ and $\xi' = \langle n', n'' \rangle$. Let the set of all expanded arcs be Ω.

We endow each expanded arc in the node-and-time expanded graph (Ξ, Ω) with a reward similar to (4.15). To derive the exact form of this reward, we need the following building blocks. For any $v, v' \in \mathscr{V}$ and $t \in \mathscr{T}$, let $M_t(v, v')$ be a J-by-J identity matrix with the $\phi(v')$-th diagonal element set equal to $1-g(v, v', t)$. We also let $\Gamma(v')$ be the $\phi(v')$-th column of the Markov transition matrix Γ.

From (4.12) and the recursive application of (4.11), we see that the probability of detection along a path $\{v_l\}_{l=0}^T$ is

$$\begin{aligned} P(\{v_l\}_{l=0}^T) = q(\phi(v_1), 1)g(v_0, v_1, 1)+ \\ q(\cdot, 1)M_1(v_0, v_1)\Gamma(v_2)g(v_1, v_2, 2)+ \\ q(\cdot, 1)M_1(v_0, v_1)\Gamma M_2(v_1, v_2)\Gamma(v_3)g(v_2, v_3, 3)+ \\ q(\cdot, 1)M_1(v_0, v_1)\Gamma M_2(v_1, v_2)\Gamma M_3(v_2, v_3)\Gamma(v_4)g(v_3, v_4, 4)+ \\ \vdots \\ q(\cdot, 1)M_1(v_0, v_1)\Gamma M_2(v_1, v_2)\Gamma M_3(v_2, v_3) \cdot \ldots \cdot \Gamma M_{T-1}(v_{T-2}, v_{T-1}) \\ \Gamma(v_T)g(v_{T-1}, v_T, T) \quad , \end{aligned} \tag{4.16}$$

which gives insight into a class of bounds on the probability of detection including the static bound $d_0(v_t, t)$. If we replace $M_t(\cdot, \cdot)$ by the identity matrix in (4.16), we find that

$$\begin{aligned} P(\{v_l\}_{l=0}^T) \le q(\phi(v_1), 1)g(v_0, v_1, 1) + \\ q(\cdot, 1)\Gamma(v_2)g(v_1, v_2, 2) + \\ q(\cdot, 1)\Gamma\Gamma(v_3)g(v_2, v_3, 3) + \\ q(\cdot, 1)\Gamma\Gamma\Gamma(v_4)g(v_3, v_4, 4) + \\ \vdots \\ q(\cdot, 1)\Gamma^{T-2}\Gamma(v_T)g(v_{T-1}, v_T, T). \end{aligned} \tag{4.17}$$

In (4.17), the "reward" received during a time period is simply the probability of detection during that time period and depends only on the current and previous vertices. Hence, it is possible to compute an upper bound on the optimal probability of detection by finding a path $\{v_l\}_{l=0}^T$ that maximizes the right-hand side in (4.17).

This calculation amounts to a longest-path problem. If we replace each $M_t(\cdot,\cdot)$ by the identity matrix everywhere except the last matrix of each line in (4.16), we obtain

$$
\begin{aligned}
P(\{v_l\}_{l=0}^T) \leq q(\phi(v_1), 1)g(v_0, v_1, 1) &+ \\
q(\cdot, 1)M_1(v_0, v_1)\Gamma(v_2)g(v_1, v_2, 2) &+ \\
q(\cdot, 1)\Gamma M_2(v_1, v_2)\Gamma(v_3)g(v_2, v_3, 3) &+ \\
q(\cdot, 1)\Gamma\Gamma M_3(v_2, v_3)\Gamma(v_4)g(v_3, v_4, 4) &+ \\
&\vdots \\
q(\cdot, 1)\Gamma^{T-2}M_{T-1}(v_{T-2}, v_{T-1})\Gamma(v_T)g(v_{T-1}, v_T, T)&.
\end{aligned}
\tag{4.18}
$$

Now, the reward received during each time period also depends on the searcher's position two time periods ago and the problem of finding a path that maximizes the right-hand side is no longer a longest-path problem. However, the bound remains valid with the following minor modification, where the maximization of a matrix with a single element different from zero or one is simply the maximization of that element:

$$
\begin{aligned}
P(\{v_l\}_{l=0}^T) \leq q(\phi(v_1), 1)g(v_0, v_1, 1) &+ \\
q(\cdot, 1)\left(\max_{v\in\mathscr{R}(v_1)} M_1(v, v_1)\right)\Gamma(v_2)g(v_1, v_2, 2) &+ \\
q(\cdot, 1)\Gamma\left(\max_{v\in\mathscr{R}(v_2)} M_2(v, v_2)\right)\Gamma(v_3)g(v_2, v_3, 3) &+ \\
q(\cdot, 1)\Gamma\Gamma\left(\max_{v\in\mathscr{R}(v_3)} M_3(v, v_3)\right)\Gamma(v_4)g(v_3, v_4, 4) &+ \\
&\vdots \\
q(\cdot, 1)\Gamma^{T-2}\left(\max_{v\in\mathscr{R}(v_{T-1})} M_{T-1}(v, v_{T-1})\right)\Gamma(v_T)g(v_{T-1}, v_T, T)&.
\end{aligned}
\tag{4.19}
$$

After this modification, we see that the reward during each time period only depends on the current and previous vertices. Hence, again, it is possible to compute an upper bound on the optimal probability of detection by solving a longest-path problem. In fact, this is exactly the approach to the static bound and it can be shown that the reward in the longest-path problem $c_{n,n'}$, see (4.15), can be deduced from (4.19). Specifically, when the current subpath in (4.15) is $\{v_0\}$, we have for arc $(n, n') = (\langle v, s\rangle, \langle v', s+1\rangle) \in \mathscr{A}$ that

$$
c_{n,n'} = q(\cdot, 1)\Gamma^{s-1}\left(\max_{v''\in\mathscr{R}(v)} M_s(v'', v)\right)\Gamma(v')g(v, v', s+1).
$$

Using similar arguments, we define the directional static bound as follows. Clearly,

$$
\begin{aligned}
P(\{v_l\}_{l=0}^{T}) \leq q(\phi(v_1), 1)g(v_0, v_1, 1) +&\\
q(\cdot, 1)M_1(v_0, v_1)\Gamma(v_2)g(v_1, v_2, 2) +&\\
q(\cdot, 1)\left(\max_{v \in \mathscr{R}(v_1)} M_1(v, v_1)\right)\Gamma M_2(v_1, v_2)\Gamma(v_3)g(v_2, v_3, 3) +&\\
q(\cdot, 1)\Gamma\left(\max_{v \in \mathscr{R}(v_2)} M_2(v, v_2)\right)\Gamma M_3(v_2, v_3)\Gamma(v_4)g(v_3, v_4, 4) +&\\
\vdots\\
q(\cdot, 1)\Gamma^{T-3}\left(\max_{v \in \mathscr{R}(v_{T-2})} M_{T-2}(v, v_{T-2})\right)\Gamma M_{T-1}(v_{T-2}, v_{T-1})\Gamma(v_T)g(v_{T-1}, v_T, T).&
\end{aligned}
\tag{4.20}
$$

Hence, we can compute an upper bound on the optimal probability of detection by finding a path $\{v_l\}_{l=0}^{T}$ that maximizes the right-hand side of (4.20). This calculation amounts to a longest-path problem in the node-and-time expanded graph (Ξ, Ω). The arc reward in this longest-path problem is deduced from (4.20). Specifically, an expanded arc $(\xi, \xi') = (\langle n'', n\rangle, \langle n, n'\rangle) \in \Omega$, with $n'' = \langle v'', s-1\rangle$, $n = \langle v, s\rangle$, and $n' = \langle v', s+1\rangle$, is endowed with the reward

$$
c_{\xi,\xi'} = q(\cdot, 1)\Gamma^{s-2}\left(\max_{v''' \in \mathscr{R}(v'')} M_{s-1}(v''', v'')\right)\Gamma M_s(v'', v)\Gamma(v')g(v, v', s+1).
$$

We refer to the node-and-time expanded graph (Ξ, Ω) with the arc rewards $c_{\xi,\xi'}$ from (4.21) as the node-and-time expanded network. Since the node-and-time expanded graph is acyclic, longest-path problems are solvable by standard shortest-path algorithms.

In view of the above discussion, we obtain the following result.

Theorem 4.1. *For any $v' \in \mathscr{V}$ and $t \in \mathscr{T}$, let*

(i) $d_0(v', t)$ be the value of the longest-path from node $\langle v', t\rangle$ to node $\hat{n}$ in the time-expanded graph $(\mathscr{N}, \mathscr{A})$ with arc rewards given by (4.20), and

(ii) $\delta_0(v, v', t)$ be the value of the longest-path from expanded node $\langle\langle v, t-1\rangle, \langle v'\ t\rangle\rangle$ to expanded node $\hat{\xi}$ in the node-and-time expanded graph (Ξ, Ω) with arc rewards given by (4.21).

Then, both $d_0(v', t)$ and $\delta_0(v, v', t)$ are upper bounds on the probability of detection during time periods $t+1, t+2, \ldots, T$ for any path $\{v_l\}_{l=0}^{T}$ with $v_{t-1} = v$ and $v_t = v'$. Moreover, $\delta_0(v, v', t) \leq d_0(v', t)$.

We refer to $\delta_0(v, v', t)$ as the directional static bound and see from Theorem 4.1 that it is at least as strong as the static bound. Clearly, building the node-and-time expanded graph (Ξ, Ω), computing the associated rewards, and calculating the longest-paths take some computing time. However, the process is only carried out once before the start of the Enhanced Branch-and-Bound Algorithm and the computed bounds are stored for later use. Hence, the time for computing the directional static bounds remains small compared to the overall run time. Empirical evidence in Sato and Royset (2010) indicates that implementing the directional static bound is beneficial.

4.2 Mathematical Programming Formulations

Although it might be conceptually possible to extend the above path-centered formulations of the previous section to multiple searchers, multiple targets, and additional constraints, it is usually easier to formulate the more general path-constrained searcher problems as mathematical programs. A significant benefit of this approach is that a large number of well-developed (general purpose) mathematical programming algorithms become available. Therefore, at least in many cases, one avoids the need for developing specialized algorithms, a significant practical benefit. Section 4.3 provides background on mathematical programming algorithms. We start, however, with formulation of path-constrained search problems as mathematical programs. We divide the exposition in two parts: first we deal with a group of identical searchers and a single target and second address the full problem with different types of searchers, many targets, and additional constraints.

4.2.1 Homogeneous Searchers and Single Target

As above, we let the search take place in a finite set of cells $\mathscr{J} = \{1, \ldots, J\}$ and over a finite set of time periods $\mathscr{T} = \{1, 2, \ldots, T\}$. We let $t = 0$ represent the time prior to search and set $\mathscr{T}_0 = \{0\} \cup \mathscr{T}$. We consider a single target and the quantity $\omega_t \in \mathscr{J}$ denotes the (random) cell that the target occupies during time $t \in \mathscr{T}$. The vector of cells $\omega = (\omega_1, \omega_2, \ldots, \omega_T)$ gives a possible path for the target and $p(\omega)$ denotes the given probability that the target takes that path. The set Ω denotes the collection of all possible paths with positive probability $p(\omega)$ and, of course, $\sum_{\omega \in \Omega} p(\omega) = 1$. In practice, Ω and $p(\omega)$ are generated using Monte Carlo sampling from (complex) target motion models (as in U.S. Coast Guard's decision aid SAROPS; see Kratzke et al. 2010) or defined implicitly by Markov transition matrices. We consider both ways of specifying target movement and refer to the former way as a *conditional target model* and, as before, to the latter way as a *Markovian target model*. Although not as general, a Markovian target model offers computational benefits as we see in Sects. 4.2.1.3 and 4.3.3.

It is trivial to extend the current framework to situations with a target that may not be present and a target that enters and leaves an area during the time horizon by adding dummy cells. However, we omit details to avoid complicating the notation.

There are S identical searchers with $z(j,0)$ searchers occupy cell j in time period 0. During each time $t \in \mathscr{T}_0$, each searcher occupies a cell or is in transit between cells. When occupying a cell j, a searcher may select to move to any cell "adjacent" to j as defined by the forward star $\mathscr{F}(j) \subset \mathscr{J}$. We also let $\mathscr{R}(j) \subset \mathscr{J}$ denote the reverse star of cell j, which represents the set of cells from which a searcher can reach cell j in one move. By convention, $j \in \mathscr{F}(j)$ and $j \in \mathscr{R}(j)$. A searcher requires $d(j,j')$ time periods to move from cell j to cell $j' \in \mathscr{F}(j)$ and to search cell j' for one time period. Since the time to search the "destination" cell j' is included in $d(j,j')$, we have that $d(j,j') \geq 1$ for all j,j' and $d(j,j') = 1$ only if the time to move from j to j' is zero. Naturally, a move can also take one or more time periods, during which the searcher is unable to search.

We let $Z(j,j',t)$ denote the number of searchers that occupy cell j in time period $t \in \mathscr{T}_0$ and that move to cell j' next, and let Z denote the vector with components $Z(j,j',t), j,j' \in \mathscr{J}$, and $t \in \mathscr{T}_0$. We refer to Z as a search plan.

We assume that every searcher is equipped with one imperfect sensor. Each time period $t \in \mathscr{T}$ in which a searcher occupies a cell, the searcher's sensor takes one "look" in the cell for the target. When a searcher is in transit between cells, the sensor is inactive. If the target and a searcher occupy cell j in time period t and j' is the searcher's previous cell, then the probability that the searcher's look during time period t detects the target is $g(j',j,t) \in [0,1)$. As before, we refer to this probability as the *glimpse-detection probability*.

We assume that the glimpse-detection probabilities are independent across all the searchers' looks. Recall that $Z(j',j,t-d(j',j))$ is the number of searchers that are present in cell j and that reached that cell from cell j'. Clearly, to arrive at cell j for time t, these searchers need to have departed cell j' at time $t-d(j',j)$. Hence, given search plan Z, the probability that no searcher detects the target in cell j in time period t, given that the target occupies cell j at that time, equals

$$\prod_{j' \in \mathscr{R}(j)} [1 - g(j',j,t)]^{Z(j',j,t-d(j',j))}$$

$$= \exp\left(-\sum_{j' \in \mathscr{R}(j)} \alpha(j',j,t) Z(j',j,t-d(j',j))\right), \tag{4.21}$$

where for all $j \in \mathscr{J}$, $j' \in \mathscr{R}(j)$, and $t \in \mathscr{T}$,

$$\alpha(j',j,t) = -\ln[1 - g(j',j,t)] \tag{4.22}$$

is the *detection rate*. The detection rate can often be determined by the models described in Chap. 2.

We seek to find a search plan Z such that the probability that no searcher detects the target is minimized subject to constraints on Z, especially those that ensure that

Z does not violate the requirement that each searcher must move according to the forward start and obey the given travel times between cells. The next subsections provide mathematical programming formulations of this problem.

4.2.1.1 Nonlinear Optimization Model

The problem of minimizing the nondetection problem takes the form of a mixed-integer nonlinear program.

Model SP1:

Indices

j, j'	cells ($j, j' \in \mathscr{J} = \{1, \ldots, J\}$).
t	time periods ($t \in \mathscr{T}_0 = \{0\} \cup \mathscr{T}$, $\mathscr{T} = \{1, \ldots, T\}$).
ω	path of target ($\omega \in \Omega$).

Sets

$\mathscr{F}(j) \subseteq \mathscr{J}$	forward star of cell j for searchers.
$\mathscr{R}(j) \subseteq \mathscr{J}$	reverse star of cell j for searchers.

Parameters

$\alpha(j', j, t)$	detection rate in cell j in time period t against the target for a searcher that previously occupied j'.
$\zeta(j, t, \omega)$	1 if cell j is on target path ω in time period t, otherwise 0.
$z(j, 0)$	number of searchers that occupy cell j in time period 0.
$p(\omega)$	probability that the target takes path ω.
$d(j, j')$	number of time periods needed for a searcher to move directly from cell j to cell j' and search j'.

Decision Variables

$Z(j, j', t)$	number of searchers that occupy cell j in time period t and that move to cell j' next. (Z denotes the vector with components $Z(j, j', t), j, j' \in \mathscr{J}, t \in \mathscr{T}_0$.)
$Y(j, t)$	number of searchers that occupy cell j in time period t. (Y denotes the vector with components $Y(j, t), j \in \mathscr{J}, t \in \mathscr{T}$.)

Function

$f(Y)$	nondetection probability of target given Y

$$= \sum_{\omega \in \Omega} p(\omega) \exp\left(- \sum_{j \in \mathscr{J}, t \in \mathscr{T}} \zeta(j, t, \omega) \alpha(j, t) Y(j, t) \right). \tag{4.23}$$

Formulation

$$\min f(Y)$$

$$\text{s.t.} \sum_{j' \in \mathscr{R}(j)} Z(j', j, t - d(j', j)) = \sum_{j' \in \mathscr{F}(j)} Z(j, j', t) \quad \forall j \in \mathscr{J}, t \in \mathscr{T} \tag{4.24}$$

$$\sum_{j' \in \mathscr{F}(j)} Z(j, j', 0) = z(j, 0) \quad \forall j \in \mathscr{J} \tag{4.25}$$

$$\sum_{j' \in \mathscr{R}(j)} Z(j', j, t - d(j', j)) = Y(j, t) \quad \forall j \in \mathscr{J}, t \in \mathscr{T} \tag{4.26}$$

$$Z(j, j', t) \geq 0 \quad \forall j, j' \in \mathscr{J}, t \in \mathscr{T}_0 \tag{4.27}$$

$$Y(j, t) \in \{0, 1, 2, \ldots, S\} \quad \forall j \in \mathscr{J}, t \in \mathscr{T} \tag{4.28}$$

The objective function f in (4.23) aims to minimize the nondetection probability and is convex and continuously differentiable. However, the restriction of $Y(j,t)$ to integers makes **SP1** a mixed-integer nonlinear program with a convex relaxation. Constraints (4.24) ensure route continuity for each searcher. Specifically, the right-hand side of (4.24) gives the total number of searchers in cell j at time t. The sum over j' accounts for all the possible cells these searchers might move to next. The left-hand side of (4.24) accounts for where these searcher came from. Each one of them must have been in cell j' immediately prior to arriving in cell j. To ensure that they arrived in cell j for time t, they departed j' at time $t - d(j', j)$. The constraints (4.25) implement the initial conditions for the searchers.

Since **SP1** is in the class of mixed-integer nonlinear programs, it is clear that general purpose solvers such as Bonmin (COIN-OR 2009), DICOPT (Grossmann et al. 2008), and even the Microsoft Excel Solver apply. In Sect. 4.3, we describe the algorithms behind such solvers. Next, we discuss two reformulations of **SP1**, which are mixed-integer *linear* programs. This enables the application of a larger collection of optimization solvers and often computational savings. We stress that there is no approximation introduced in these reformulations. The first linearization is applicable in the case of a conditional target model with a moderate number of possible target paths. The second linearization is limited to the situation with a Markovian target model.

4.2.1.2 Linearization for Conditional Target Model

The objective function in **SP1** is a finite sum of exponential functions over all possible target paths; see (4.23). We here make the additional assumption that the detection rate is constant over all cells and time periods and simply write α. Then, each exponential function has as argument an integer multiple of α between 0 and ST, where S is the number of searchers. Hence, the objective function in **SP1** can equivalently be represented by a piecewise linear functions, with a finite number of pieces. This observation leads to the first linearization of **SP1**, where some indices, sets, parameters, and variables are as in **SP1**.

Model SP1-L:

Additional Indices

i number of looks on a target path ($i = 0, 1, \ldots, ST$).

Additional Variables

$U(\omega)$ auxiliary variable representing nondetection probability given target path ω.

Formulation

$$\min \sum_{\omega \in \Omega} p(\omega) U(\omega)$$

$$\text{s.t. } e^{-i\alpha}(1 + i - ie^{-\alpha}) + \frac{1}{\alpha} e^{-i\alpha}(e^{-\alpha} - 1) \sum_{j \in \mathscr{J}, t \in \mathscr{T}} \zeta(j, t, \omega) \alpha Y(j, t) \leq U(\omega) \quad \forall\, \omega, i \tag{4.29}$$

and (4.24)–(4.28)

SP1-L is a mixed-integer linear program. Constraints (4.29) ensure that the optimal solution results in a value of $U(\omega)$ that is exactly the conditional nondetection probability given that the target follows path ω. Specifically,

$$\exp\left(- \sum_{j \in \mathscr{J}, t \in \mathscr{T}} \zeta(j, t, \omega) \alpha Y(j, t) \right) \tag{4.30}$$

is simply a function of the form $\exp(-\alpha z)$, with $z = \sum_{j \in \mathscr{J}, t \in \mathscr{T}} \zeta(j, t, \omega) Y(j, t)$. Since $\zeta(j, t, \omega)$ is binary and $Y(j, t)$ is a nonnegative integer no larger than S, z can only take on a finite number of values. In fact, it suffices to consider z between 0 and ST. Noninteger values of z are immaterial. Hence, we can replace $\exp(-\alpha z)$ by the piecewise linear function that coincides with $\exp(-\alpha z)$ for $z = 0, 1, \ldots, ST$. Each affine piece in this function is of the form given on the left-hand side of (4.29). Using a standard technique for converting a piecewise linear function into a collection of affine constraints leads to (4.29).

The other constraints in **SP1-L** are identical to those in **SP1**. The number of constraints and variables in **SP1-L** grows linearly in the number of possible target paths and, hence, the formulation may become difficult to solve for large numbers of such paths. This motivates a second linearization of **SP1**.

4.2.1.3 Linearization for Markovian Target Model

The second linearization of **SP1**, denoted by **SP1-LM**, assumes a Markovian target model where the target at time $t \in \mathscr{T}$ moves according to a transition probability matrix Γ_t with elements $\Gamma_t(j,j'), j,j' \in \mathscr{J}$. Specifically, $\Gamma_t(j,j')$ is the probability that a target occupying cell j in time period t occupies cell j' in time period $t+1$. As we see below, it is not necessary to enumerate all possible target paths in the case of a Markovian target model.

We derive **SP1-LM** from **SP1** by introducing an "information state" $Q(j,t)$ which equals the probability that the target occupies cell j in time period t and that the target has not been detected prior to t. Given this information state and a search plan with s searchers occupying cell j in time period t, the probability of detection in cell j in time period t and no prior detection, is simply $Q(j,t)(1-\exp[-s\alpha(j,t)])$, where $\alpha(j,t)$ is the detection rate of each searcher in cell j in time period t. Suppose that a search plan is described by the binary variables

$$V(j,t,s) = 1 \text{ if } s \text{ searchers occupy cell } j \text{ in time period } t \text{, and 0 otherwise.} \quad (4.31)$$

Then, the probability of detection over the full time horizon becomes

$$\sum_{t\in\mathscr{T}}\sum_{j\in\mathscr{J}} Q(j,t)\left(1-\exp\left[-\alpha(j,t)\sum_{s=1}^{S} sV(j,t,s)\right]\right). \quad (4.32)$$

The information state $Q(j,t)$ depends on the search plan as follows. Clearly, $Q(j,1) = p(j,1)$, the given probability that the target occupies cell j initially. Moreover, it follows from the definition of $Q(j,t)$ and the assumption of a Markovian target model that

$$Q(j,t+1) = \sum_{j'\in\mathscr{J}} \Gamma_t(j',j)Q(j',t)\exp\left[-\alpha(j',t)\sum_{s=1}^{S} sV(j',t,s)\right] \quad (4.33)$$

for all $j \in \mathscr{J}$ and $t = 1,2,\ldots,T-1$. While the expressions (4.32) and (4.33) are nonlinear, they can be linearized as shown in the following formulation; see further explanation below the model statement.

Model SP1-LM:

Additional Indices

s	number of searchers in a cell ($s \in \mathscr{S} = \{1,\ldots,S\}$).

Additional Parameters

$\alpha(j,t)$	detection rate in cell j in time period t for any searcher.
$\Gamma_t(j,j')$	probability that a target that occupies cell j in time period t occupies cell j' in time period $t+1$.

$p(j,t)$	probability that the target occupies cell j in time period t, i.e., $p(j,t) = \sum_{j'} p(j', t-1)\Gamma_{t-1}(j', j)$, $t = 2, 3, \ldots, T$, $p(j,1)$ given.

Additional Variables

$Q(j,t)$	probability that the target occupies cell j in time period t and target not detected prior to t.
$R(j,t,s)$	auxiliary variable that equals $Q(j,t)(1-e^{-s\alpha(j,t)})$ if $V(j,t,s)=1$ and otherwise 0.
$V(j,t,s)$	1 if there are s searchers that occupy cell j in time period t and otherwise 0.
$W(j,t)$	auxiliary variable that equals $Q(j,t)e^{-s\alpha(j,t)}$ if $V(j,t,s) = 1$ and otherwise $Q(j,t)$.

Formulation

$$\min\ 1 - \sum_{t\in\mathscr{T}} \sum_{j\in\mathscr{J}} \sum_{s\in\mathscr{S}} R(j,t,s) \tag{4.34}$$

$$\text{s.t. } R(j,t,s) \le p(j,t)(1-e^{-s\alpha(j,t)})V(j,t,s) \quad \forall\, j \in \mathscr{J}, t \in \mathscr{T}, s \in \mathscr{S} \tag{4.35}$$

$$R(j,t,s) \le (1-e^{-s\alpha(j,t)})Q(j,t) \quad \forall\, j \in \mathscr{J}, t \in \mathscr{T}, s \in \mathscr{S} \tag{4.36}$$

$$Q(j,t+1) = \sum_{j'\in\mathscr{J}} \Gamma_t(j',j)W(j',t) \quad \forall\, j \in \mathscr{J}, t=1,\ldots,T-1 \tag{4.37}$$

$$W(j,t) \le Q(j,t) \quad \forall\, j \in \mathscr{J}, t \in \mathscr{T} \tag{4.38}$$

$$W(j,t) \le e^{-s\alpha(j,t)}Q(j,t) + p(j,t(1-e^{-s\alpha(j,t)})(1-V(j,t,s)) \quad \forall\, j \in \mathscr{J}, t \in \mathscr{T}, s \in \mathscr{S} \tag{4.39}$$

$$Q(j,1) = p(j) \quad \forall\, j \in \mathscr{J} \tag{4.40}$$

$$Q(j,t) \le p(j,t) \quad \forall\, j \in \mathscr{J}, t \in \mathscr{T}_0 \tag{4.41}$$

$$\sum_{j'\in\mathscr{R}(j)} Z(j',j,t-d(j',j)) = \sum_{s} sV(j,t,s) \quad \forall\, j \in \mathscr{J}, t \in \mathscr{T} \tag{4.42}$$

$$\sum_{s\in\mathscr{S}} V(j,t,s) \le 1 \quad \forall\, j \in \mathscr{J}, t \in \mathscr{T} \tag{4.43}$$

(4.24), (4.25)

$$Q(j,t), R(j,t,s), W(j,t), Z(j,j',t) \ge 0 \quad \forall\, j,j' \in \mathscr{J}, t \in \mathscr{T}, s \in \mathscr{S} \tag{4.44}$$

$$V(j,t,s) \in \{0,1\} \quad \forall\, j \in \mathscr{J}, t \in \mathscr{T}, s \in \mathscr{S} \tag{4.45}$$

The objective function (4.34) in **SP1-LM** gives the probability of nondetection; its correctness follows from (4.32). However, since (4.32) is nonlinear, we linearize it using the auxiliary variable $R(j,t,s)$, which equals $Q(j,t)(1-e^{-s\alpha(j,t)})$ if $V(j,t,s) = 1$ and equals 0 otherwise. This linearization is accomplished using constraints (4.35) and (4.36). This is a "big-M" type of formulation (see Rardin 1997, pp. 642–643) where any constant at least as large as $Q(j,t)$ would suffice in front of $(1-e^{-s\alpha(j,t)})$ in (4.35). Recall that $Q(j,t)$ is the probability that the target occupies cell j in time period t *and* target not detected prior to t. Moreover, recall that $p(j,t)$ is the probability that the target occupies cell j in time period t. Hence, $p(j,t) \geq Q(j,t)$ for all j,t. Consequently, we set the "big-M" in (4.35) to $p(j,t)$. We also use $p(j,t)$ to bound the range of $Q(j,t)$ in (4.41).

The evolution of the information state is also nonlinear; see (4.33). In **SP1-LM**, we linearize that expression by means of the auxiliary variable $W(j,t)$ and constraints (4.37)–(4.39). Note that $W(j,t)$ equals $Q(j,t)e^{-s\alpha(j,t)}$ if $V(j,t,s) = 1$ and equals $Q(j,t)$ otherwise. The initial target location is accounted for in (4.40). The binary variable $V(j,t,s)$ relates to $Z(j,j',t)$ in (4.42) and (4.43).

4.2.2 Heterogenous Searchers and Multiple Targets

We next extend the above formulation of **SP1** to the case of multiple searchers of different types, multiple targets, additional constraints. We start with additional notation.

There are K independent targets present with each target $k \in \mathscr{K} = \{1,2,\ldots,K\}$ occupying one cell in each time period. The quantity $\omega_{k,t} \in \mathscr{J}$ denotes the (random) cell that target k occupies during time $t \in \mathscr{T}$. The vector of cells $\omega_k = (\omega_{k,1}, \omega_{k,2}, \ldots, \omega_{k,T})$ denotes a possible path for target k and $p_k(\omega_k)$ denotes the given probability that target k takes that path. The set Ω_k denotes the collection of all possible paths for target k with positive probability $p_k(\omega_k)$ and, of course, $\sum_{\omega_k \in \Omega_k} p_k(\omega_k) = 1$ for all k.

There are L classes of searchers with each class $l \in \mathscr{L} = \{1,2,\ldots,L\}$ containing S_l identical searchers. During each time $t \in \mathscr{T}_0$, each searcher occupies a cell or is in transit between cells. When occupying a cell j, a searcher of class l may select to move to any cell "adjacent" to j as defined by the forward star $\mathscr{F}_l(j) \subset \mathscr{J}$. We also let $\mathscr{R}_l(j) \subset \mathscr{J}$ denote the reverse star of cell j, which represents the set of cells from which a searcher of class l can reach cell j in one move. By convention, $j \in \mathscr{F}_l(j)$ and $j \in \mathscr{R}_l(j)$. A searcher of class l requires $d_l(j,j')$ time periods to move from cell j to cell $j' \in \mathscr{F}_l(j)$ and to search cell j' for one time period. Since the time to search the "destination" cell j' is included in $d_l(j,j')$, we have that $d_l(j,j') \geq 1$ for all l,j,j' and $d_l(j,j') = 1$ only if the time to move from j to j' is zero. Naturally, a move can also take one or more time periods, during which the searcher is unable to search.

We let $Z_l(j, j', t)$ denote the number of searchers of class l that occupy cell j in time period $t \in \mathscr{T}_0$ and that move to cell j' next, and let Z denote the vector with components $Z_l(j, j', t)$, $l \in \mathscr{L}$, $j, j' \in \mathscr{J}$, and $t \in \mathscr{T}_0$. We refer to Z as a search plan.

The probability that one look for a target by a searcher in a cell detects the target, given that the target currently occupies the cell, may depend on the searcher class (is it a high- or low-quality searcher?), the target's characteristic (is it shiny or camouflaged?), the cell (is it forested or open?) and time of day (is it bright mid-day or dark midnight?). Specifically, if target k and a searcher of class l occupy cell j in time period t and j' is the searcher's previous cell, then the probability that the searcher's look during time period t detects the target is $g_{l,k}(j', j, t) \in [0, 1)$. As above, we refer to this probability as the *glimpse-detection probability*.

We retain the assumption that the glimpse-detection probabilities are independent across all the searchers' looks. Hence, given search plan Z, the probability that no searcher detects target k in cell j in time period t, given that target k occupies cell j at that time, equals

$$\prod_{l \in \mathscr{L}} \prod_{j' \in \mathscr{R}_l(j)} [1 - g_{l,k}(j', j, t)]^{Z_l(j', j, t - d_l(j', j))}$$

$$= \exp\left(- \sum_{l \in \mathscr{L}} \sum_{j' \in \mathscr{R}_l(j)} \alpha_{l,k}(j', j, t) Z(l, j', j, t - d_l(j', j)) \right), \tag{4.46}$$

where for all $l \in \mathscr{L}$, $j \in \mathscr{J}$, $j' \in \mathscr{R}_l(j)$, $t \in \mathscr{T}$, and $k \in \mathscr{K}$,

$$\alpha_{l,k}(j', j, t) = -\ln[1 - g_{l,k}(j', j, t)] \tag{4.47}$$

is the *detection rate*.

We seek to minimize, by choice of a search plan Z, the probability of not detecting the target with the *largest* nondetection probability during the time horizon. Another possibility is to minimize the *sum* of the nondetection probabilities across the targets. This implies only minor changes to the below formulation. The latter choice places less emphasis on the "least detectable" target and usually distributes search effort more evenly across the targets. The former choice, however, ensures that no target is "ignored" with a corresponding low detection probability.

The choice of search plan is subject to the route constraints induced by the forward and reverse stars $\mathscr{F}_l(j)$ and $\mathscr{R}_l(j)$, the given initial condition that $z_l(j, 0)$ searchers of class l occupy cell j in time period 0, and *deconflication* constraints related to the maximum number of searchers that can occupy a cell at any time and other operational requirements. We let n_j be the maximum number of searchers allowed to occupy cell j during any one time period $t \in \mathscr{T}$. Moreover, for each possible move between two cells for a searcher, we define a corresponding set of incompatible moves between cells that would cause interference if carried out by another searcher. Specifically, if a searcher of class l moves from cell j to cell j'

starting in time period t, then the set $\mathscr{D}(l, j, j', t)$ gives all quadruples of searcher classes, cell pairs, and time periods that are incompatible with that searcher's move.

One can easily incorporate essentially any other conceivable constraint with little difficulty. Though, these additions might result in additional computing times. We omit further details and simply reference Rardin (1997) for an accessible introduction to formulation of mathematical programs and to Sato and Royset (2010) for possible types of additional constraints related to risk and fuel consumption.

The formulation then take the form of a mixed-integer nonlinear program.

Model SPX:

Indices

j, j'	cells ($j, j' \in \mathscr{J} = \{1, \ldots, J\}$).
t	time periods ($t \in \mathscr{T}_0 = \{0\} \cup \mathscr{T}$, $\mathscr{T} = \{1, \ldots, T\}$).
l	searcher class ($l \in \mathscr{L} = \{1, \ldots, L\}$).
k	target ($k \in \mathscr{K} = \{1, \ldots, K\}$).
ω_k	path of target k ($\omega_k \in \Omega_k$).

Sets

$\mathscr{F}_l(j) \subseteq \mathscr{J}$	forward star of cell j for searcher of class l.
$\mathscr{R}_l(j) \subseteq \mathscr{J}$	reverse star of cell j for searcher of class l.
$\mathscr{D}(l, j, j', t)$	set of quadruples (l', j'', j''', t') incompatible with a searcher of class l that moves from j to j' starting in time period t.

Parameters

$\alpha_{l,k}(j', j, t)$	detection rate in cell j in time period t against target k for a searcher of class l when the searcher previously occupied j'.
$\zeta(j, t, \omega_k)$	1 if cell j is on target path ω_k in time period t, otherwise 0.
$z_l(j, 0)$	number of searchers of class l that occupy cell j in time period 0.
S_l	number of searchers of class l.
$p_k(\omega_k)$	probability that target k takes path ω_k.
$d_l(j, j')$	number of time periods needed for a searcher of class l to move directly from cell j to cell j' and search j'.
n_j	maximum number of searchers that occupy cell j in a time period.

Decision Variables

$Z_l(j, j', t)$ number of searchers of class l that occupy cell j in time period t and that move to cell j' next. (Z denotes the vector with components $Z_l(j, j', t)$, $l \in \mathscr{L}, j, j' \in \mathscr{J}, t \in \mathscr{T}_0$.)

$Y_k(j, t)$ auxiliary variable representing total detection rate applied to cell j against target k in time period t. (Y_k denotes the vector with components $Y_k(j, t)$, $j \in \mathscr{J}, t \in \mathscr{T}$.)

Functions

$f_k(Y_k)$ nondetection probability of target k

$$= \sum_{\omega_k \in \Omega_k} p_k(\omega_k) \exp\left(- \sum_{j \in \mathscr{J}, t \in \mathscr{T}} \zeta(j, t, \omega_k) Y_k(j, t) \right). \tag{4.48}$$

Formulation

$$\min \max_{k \in \mathscr{K}} f_k(Y_k) \tag{4.49}$$

$$\text{s.t.} \sum_{j' \in \mathscr{R}_l(j)} Z_l(j', j, t - d_l(j', j)) = \sum_{j' \in \mathscr{F}_l(j)} Z_l(j, j', t) \quad \forall\, l \in \mathscr{L}, j \in \mathscr{J}, t \in \mathscr{T} \tag{4.50}$$

$$\sum_{j' \in \mathscr{F}_l(j)} Z_l(j, j', 0) = z_l(j, 0) \quad \forall\, l \in \mathscr{L}, j \in \mathscr{J} \tag{4.51}$$

$$\sum_{l \in \mathscr{L}} \sum_{j' \in \mathscr{R}_l(j)} \alpha_{l,k}(j', j, t) Z_l(j', j, t - d_l(j', j)) = Y_k(j, t) \quad \forall\, j \in \mathscr{J}, t \in \mathscr{T}, k \in \mathscr{K} \tag{4.52}$$

$$\sum_{l \in \mathscr{L}} \sum_{j' \in \mathscr{R}_l(j)} Z_l(j', j, t - d_l(j', j)) \le n_j \quad \forall\, j \in \mathscr{J}, t \in \mathscr{T} \tag{4.53}$$

$$Z_l(j, j', t) + Z_{l'}(j'', j''', t') \le 1 \quad \forall\, l \in \mathscr{L}, j, j' \in \mathscr{F}_l(j), \quad t \in \mathscr{T}_0, (l', j'', j''', t') \in \mathscr{D}(l, j, j', t) \tag{4.54}$$

$$Z_l(j, j', t) \in \{0, 1, 2, \ldots, S_l\} \quad \forall\, l \in \mathscr{L}, j, j' \in \mathscr{J}, t \in \mathscr{T}_0 \tag{4.55}$$

$$Y_k(j, t) \ge 0 \quad \forall\, j \in \mathscr{J}, t \in \mathscr{T}, k \in \mathscr{K}. \tag{4.56}$$

The decision variable $Y_k(j, t)$ could be eliminated by substitution using (4.52), but is included for notational simplicity. We obtain the nondetection probability for target k in (4.48) from (4.46) by application of the total probability theorem and the fact that detection in cell j in time period t can occur only if the target occupies that cell at that time. The objective function (4.49) aims to minimize the largest nondetection probability. The objective function of **SPX** is convex and the nondetection probabilities $f_k(Y_k), k \in \mathscr{K}$, are convex and continuously differentiable.

Constraints (4.50) and (4.51) ensure route continuity for each searcher and initial conditions as in **SP1**. Deconfliction constraints (4.53) and (4.54) limit the number of searchers that can occupy cell j to at most n_j in any time period $t \in \mathscr{T}$ and exclude moves in conflict with each other, respectively.

We observe that **SPX** prescribes the "best" search plan prior to detection of the first target. In the presence of multiple targets, one might want to explicitly account for events after the first detection and possibly deviate from the plan stipulated by **SPX**. However, this leads to stochastic dynamic programs, which are computationally extremely expensive to solve and are beyond the scope of this text. The present objective of minimizing the probability of not detecting the target with the largest nondetection probability during the time horizon is a reasonable surrogate. Another tractable surrogate could be to minimize a nonnegatively weighted sum of the nondetection probabilities. Of course, in the case of a single target this complication evaporates.

Since **SPX** is in the class of mixed-integer nonlinear programs, it is clear that we still can apply general purpose solvers such as Bonmin (COIN-OR 2009), DICOPT (Grossmann et al. 2008), and the Microsoft Excel Solver.

Conceptually, it is possible to linearize **SPX** in a similar manner as described above for **SP1**. **SP1-L**, which linearizes **SP1** in the case of a conditional target model, generalizes easily to a linear model equivalent to **SPX** if all detection rates $\alpha_{l,k}(j', j, t)$ in **SPX** are rational numbers. In that case, all detection rates **SPX** can be expressed as an integer multiple of a number, say, α. Hence, $Y_k(j,t)$ could be expressed as α times an auxiliary integer variable. Similar to the approach leading to **SP1-L**, the exponential terms in $f_k(Y_k)$ could then be expressed by piecewise-linear functions. Standard techniques for linearizing piecewise-linear functions would then lead to a mixed-integer linear program. If $\alpha_{l,k}(j', j, t)$ differs substantially across different elements of $\mathscr{L}$, $\mathscr{J}$, $\mathscr{T}$, and $\mathscr{K}$, α would need to be relatively small. Hence, the piecewise-linear functions may involve a large number of pieces and the resulting mixed-integer linear program may be large.

Under the same assumption on the detection rates and given a Markovian target model, **SP1-LM** generalizes to a linear model equivalent to **SPX** through a redefinition of s. While s gives the number of searchers occupying a cell during one time period in **SP1-LM**, the new linear model would require s to represent $\alpha^{\text{total}}/\alpha$, where α^{total} denotes the sum of the detection rates of all searchers occupying a cell in a time period. This sum may be larger than the number of searchers occupying the cell as each searcher would have a detection rate of $\xi\alpha$, where ξ is a positive integer. Since this linearization effectively assigns a binary variable to each possible value of the total detection rate applied to a cell, the resulting mixed-integer linear program may become large.

In view of the above discussion, we see that linearizations of **SPX** tend to be of reasonable size and practical value when all detection rates $\alpha_{l,k}(j', j, t)$ can be expressed as small integer multiples of α. For example, this is the case when all detection rates equals $1 \cdot \alpha$ for some $\alpha > 0$.

4.3 Mathematical Programming Algorithms for Path Optimization

Optimization solvers for (convex) mixed-integer nonlinear programs such as **SPX** rely on the principle of *branch-and-bound* and/or *cutting planes*. The first approach is based on the observation that if the integer variables are allowed to take on real values or are fixed to specific values, then the resulting optimization model is convex. Since convex optimization models are usually easily solved by well-developed algorithms, branch-and-bound algorithms for mixed-integer nonlinear programs solve a sequence of such convex models utilizing intermediate results to guide selection of which integer variables to fix and which ones to allow to be real valued. The expectation is that not all combinations of possible integer values for the variables need to be examined before a (near-)optimal solution is found.

Solvers relying on cutting planes consider sequences of mixed-integer linear programs that are obtained from the original nonlinear model through linear approximations, called cutting planes, of nonlinear functions. Numerical results in Royset and Sato (2010) indicate that approaches based on cutting planes tend to be computationally superior in the case of two or more searchers. For a single searcher, certain branch-and-bound algorithms are typically faster, but only if implemented with care as described in Sect. 4.1. We therefore focus on cutting planes methods here.

4.3.1 Cutting Plane Methods

The standard cutting-plane algorithm for convex (mixed-integer) programs sequentially builds and minimizes successively better piecewise-linear approximations of a convex function. The linear approximations are constructed using first-order Taylor expansion of nonlinear functions. In the case of **SPX**, the nondetection probabilities

$$f_k(Y_k) = \sum_{\omega_k \in \Omega_k} p_k(\omega_k) \exp\left(- \sum_{j \in \mathscr{J}, t \in \mathscr{T}} \zeta(j, t, \omega_k) Y_k(j, t) \right) \tag{4.57}$$

are the only nonlinear functions. We need the partial derivatives

$$\frac{\partial f_k(Y_k)}{\partial Y_k(j,t)} = -\sum_{\omega_k \in \Omega_k} p_k(\omega_k)\zeta(j,t,\omega_k)\exp\left(-\sum_{j' \in \mathscr{J}, t' \in \mathscr{T}} \zeta(j',t',\omega_k)Y_k(j',t')\right). \tag{4.58}$$

The collection of such partial derivatives comprise the gradient $\nabla f_k(Y_k)$. A cutting plane algorithm then takes the following form.

Cutting Plane Algorithm (Obtains near-optimal solutions of **SPX**)

Data. Relative optimality tolerances $\delta, \delta_I \geq 0, I = 0, 1, 2, \ldots$.

Step 0. Set the lower bound, $\underline{\xi}$, on the optimal value of **SPX** to 0; set the upper bound, $\overline{\xi}$, on the optimal value of **SPX** to 1; and set $I = 1$ and $Y^1 = 0$.

Step 1. For each k, calculate $f_k(Y_k^I)$ and $\nabla f_k(Y_k^I)$. If $\max_k f_k(Y_k^I) < \overline{\xi}$, then set $\overline{\xi} = \max_k f_k(Y_k^I)$.

Step 2. If $\overline{\xi} - \underline{\xi} \leq \delta\underline{\xi}$, then stop.

Step 3. Solve

$$\begin{aligned} \mathbf{P}_I: \quad & \min \xi \\ \text{s.t.}\ & f_k(Y_k^i) + \nabla f_k(Y_k^i)^\top (Y_k - Y_k^i) \leq \xi \quad \forall\, k \in \mathscr{K}, i = 1, 2, \ldots, I \\ & (4.50)\text{–}(4.56) \end{aligned} \tag{4.59}$$

to near optimality. That is, determine a lower bound $\underline{\xi}^{I+1}$ and a feasible solution $(\overline{\xi}^{I+1}, Y^{I+1}, Z^{I+1})$ of $\mathbf{P}_I$ such that $\overline{\xi}^{I+1} - \underline{\xi}^{I+1} \leq \delta_I \underline{\xi}^{I+1}$.

Step 4. If $\underline{\xi}^{I+1} > \underline{\xi}$, then set $\underline{\xi} = \underline{\xi}^{I+1}$.

Step 5. If $\overline{\xi} - \underline{\xi} \leq \delta\underline{\xi}$, then stop. Else, replace I by $I + 1$, and go to Step 1.

We note that Step 3 involves solving a mixed-integer linear program, for which there are efficient and robust solvers. The Cutting Plane Algorithm is guaranteed to solve **SPX** to optimality if $\delta = 0$ and $\delta_I = 0$ for all I. Fixing $\delta_I > 0$ (i.e., accepting near-optimal solutions of $\mathbf{P}_I$) does not guarantee convergence, but does often improve computational speed.

4.3.2 Cutting Plane Refinement

The cutting plane (4.59) can be strengthened whenever $Y_k(j,t)$ is an integer multiple of a real number that is common across all j and t. We describe the possibility in the context of **SP1**, with detection rates that are all α. In **SP1**, the nonlinearity is associated with $f(Y)$ and we therefore focus on that function. In particular, we take advantage of the special structure of $f(Y)$ and the integrality of $Y(j,t)$ to construct an improved linearization of $f(Y)$. The strengthened "cut" uses finite differences of the objective function $f(Y)$ by considering the perturbation from $Y(j,t)$ to $Y(j,t)+1$ while keeping all other variables fixed. Theorem 4.2 formalizes this discussion, using $\Delta(j,t)$ to denote a ST-dimensional binary vector in which the (j,t)-component is 1 and the other components are all 0.

Theorem 4.2. *For any ST-dimensional nonnegative integer vectors Y and $\hat{Y}$,*

$$f(\hat{Y}) + \sum_{j\in\mathscr{J},t\in\mathscr{T}} [f(\hat{Y} + \varDelta(j,t)) - f(\hat{Y})][Y(j,t) - \hat{Y}(j,t)] \le f(Y). \tag{4.60}$$

Proof. Let a^ω be a ST-dimensional vector defined by components $\zeta(j,t,\omega)\alpha$ and $b^\omega = -\ln p(\omega)$. Then, $a^\omega \ge 0$ and $b^\omega \ge 0$. Hence, $f(Y) = \sum_\omega f_\omega(Y)$, where $f_\omega(Y) = \exp(-a^\omega Y - b^\omega)$, and the result holds if $f_\omega(\hat{Y}) + \sum_{j\in\mathscr{J},t\in\mathscr{T}}(f_\omega(\hat{Y}+\Delta(j,t)) - f_\omega(\hat{Y}))(Y(j,t) - \hat{Y}(j,t)) \le f_\omega(Y)$ for all ω. Consequently, we need to show that $f_\omega(\hat{Y})[1 + \sum_{j\in\mathscr{J},t\in\mathscr{T}}(\exp(-\alpha^\omega(j,t)) - 1)(Y(j,t) - \hat{Y}(j,t)) - \exp(-a^\omega(Y - \hat{Y}))] \le 0$ for an arbitrary target path $\omega \in \varOmega$. Let $\beta = \exp(-\alpha)$, and let $\mathscr{N}$ denote the set of the cell-time pairs $(j,t) \in \mathscr{J} \times \mathscr{T}$ such that $\zeta(j,t,\omega) = 1$ (i.e., such that cell j is on path ω in time period t). Now, we only need to show that $\phi(\beta) = (1-\beta)k + \beta^k \ge 1$, where $k = \sum_{(j,t)\in\mathscr{N}}(Y(j,t) - \hat{Y}(j,t))$. We find that $d\phi(\beta)/d\beta = 0$ for $\beta = 1$. Hence, it follows from convexity of $\phi(\cdot)$ on $(0,\infty)$ that $\phi(\cdot)$ has a minimum value of 1 for any k.

We refer to (4.60) as a secant cut, which can then be utilized in the Cutting Plane Algorithm. Specifically, one replace (4.59) by

$$f(Y^i) + \sum_{j\in\mathscr{J},t\in\mathscr{T}} (f(Y^i + \Delta(j,t)) - f(Y^i))(Y(j,t) - Y^i(j,t)) \le \xi \;\; \forall\, i = 1,2,\ldots,I, \tag{4.61}$$

together with the other simplifications of **SP1** as compared to **SPX**. We refer to Royset and Sato (2010) for further details, and also empirical evidence that secant cuts reduce the computing times with a factor of 0.5. As an example, on a low-end laptop (of 2010), the Cutting Plane Algorithm with secant cuts obtains near-optimal solutions of **SP1**, with 5 % relative optimality gap, in less than 15 min of calculation time for problem instances with three searchers, 81 cells, and 10 time periods.

4.3.3 Cutting Plane Calculations for Markovian Target Model

A cutting-plane algorithm requires efficient means for evaluating $f_k(Y_k)$, $\nabla f_k(Y_k)$, as well as the finite difference in (4.61). Clearly, if the cardinality of $\varOmega_k$ becomes high, for example millions of paths, the computational cost associated with evaluating these expressions becomes significant and possibly prohibitively high. However, in the case of a Markovian Target Model the computations can be carried out efficiently as described next. To keep the exposition simple, we focus on the setting of **SP1**, with a common detection rate α.

Given a search plan Y, let $\rho_Y(j,t)$ be the probability that the target occupies cell j in time period t and that it is not detected in time periods $1, 2, \ldots, t-1$, and let $\sigma_Y(j,t)$ denote the probability that the target is not detected in time periods $t+1, t+2, \ldots, T$ given that it occupies cell j in time period t. Let $\rho_Y(t) = [\rho_Y(1,t), \rho_Y(2,t), \ldots, \rho_Y(J,t)]$, and let $\sigma_Y(t) = [\sigma_Y(1,t), \sigma_Y(2,t), \ldots, \sigma_Y(J,t)]$. We define $\rho_Y(j,1) = p(j,1)$, the probability that the target is cell j at time 1, and $\sigma_Y(j,T) = 1$ for any cell $j \in \mathscr{J}$. Thus, $\rho_Y(t)$ and $\sigma_Y(t)$ may be calculated recursively by

$$\rho_Y(t) = [\rho_Y(1,t-1)\exp(-\alpha Y(1,t-1)), \ldots, \rho_Y(J,t-1)\exp(-\alpha Y(J,t-1))]\Gamma_{t-1}, \tag{4.62}$$

and

$$\sigma_Y(t) = [\sigma_Y(1,t+1)\exp(-\alpha Y(1,t+1)), \ldots, \sigma_Y(J,t+1)\exp(-\alpha Y(J,t+1))]\Gamma_t^{\top}, \tag{4.63}$$

where Γ_t is the transition matrix of the Markovian target model. In this notation, for any $t \in \mathscr{T}$, we find that

$$f(Y) = \sum_{j \in \mathscr{J}} \rho_Y(j,t)\exp(-\alpha Y(j,t))\sigma_Y(j,t), \tag{4.64}$$

and components of $\nabla f(Y)$ are

$$\frac{\partial f(Y)}{\partial Y(j,t)} = -\alpha \rho_Y(j,t)\exp(-\alpha Y(j,t))\sigma_Y(j,t). \tag{4.65}$$

The calculation of finite differences $f(Y + \Delta(j,t)) - f(Y)$ follows similarly:

$$\begin{aligned}
&f(Y + \Delta(j,t)) - f(Y) \\
&= \sum_{j' \in \mathscr{J}} \rho_Y(j',t)[\exp(-\alpha Y(j',t) - \alpha\Delta(j,t)) - \exp(-\alpha Y(j',t))]\sigma_Y(j',t) \\
&= \rho_Y(j,t)[\exp(-\alpha(Y(j,t)+1)) - \exp(-\alpha Y(j,t))]\sigma_Y(j,t).
\end{aligned} \tag{4.66}$$

Thus, $f(Y)$ and its gradient and finite difference can be evaluated with moderate computational effort.

4.4 Example: Search for Four Targets

We illustrate **SPX** with a problem instances with $J = 81$ cells; see Fig. 4.1. We consider four targets that follow Markovian target models. At time period one, one target occupies each of the cells 5, 15, 20, and 66 (marked with diamonds in Fig. 4.1; cells are numbered left-to-right and from top-to-bottom). After each time period, a target remains in its current cell or moves to a cell directly above, below, left, or right of the current cell if such a cell exists. The probabilities of a target remaining in a cell from one time period to the next is 0.4, 0.3, 0.2, and 0.1, respectively, for the four targets; the probability of moving to any of the other allowable cells is equal. Hence, the target that initially occupies cell 5 moves slowly, the target that initially occupies cell 66 moves quickly, and the other two targets move at intermediate speeds.

We consider two classes of airborne searchers and set the travel time $d_l(j, j') = \max\{1, \text{round}(\delta(j, j')/\nu_l)\}$, where round$(a)$ is the nearest integer to a, $\delta(j, j')$ is the distance between j and j' measured as the Euclidean norm between the centers of the cells, and ν_l is the speed of searchers of class l; $\nu_1 = 1$, $\nu_2 = 2$ cells per time period. Moreover, we let the forward stars $\mathscr{F}_l(j) = \mathscr{F}^1(j) \cup \mathscr{F}_l^2(j)$, $l = 1, 2$, where $\mathscr{F}^1(j)$ equals the set consisting of j and the four cells sharing a side with j, if they exist, and $\mathscr{F}_l^2(j)$ equals the set of all cells j' with $d_l(j, j') \in [3, 5]$, if they exist. Hence, a searcher can after a time period either proceed and search "locally" (i.e., select a cell in $\mathscr{F}^1(j)$) or transit for several time periods to a distant cell (i.e., select a cell in $\mathscr{F}_l^2(j)$). The reverse stars $\mathscr{R}_l(j)$, $l = 1, 2$, are defined similarly.

We consider three scenarios with variable glimpse-detection probability and number of searchers as summarized in Table 4.1. In scenario 1, two searchers of class 1 occupy cell 1 in time period 0 and one searcher of class 2 initially occupies cell 81. The glimpse-detection probability of a searcher of the first class is 0.50

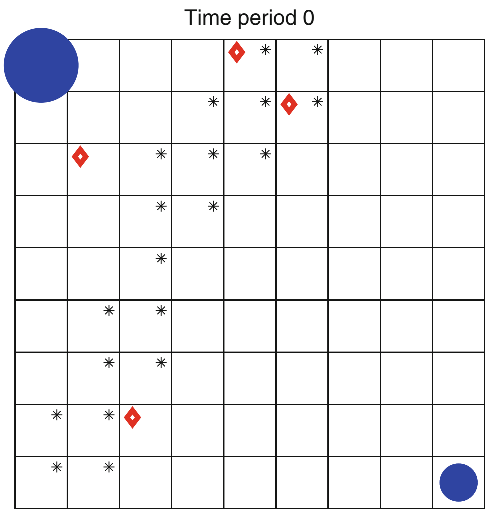


Fig. 4.1 Area of interest with *diamonds* indicating initial location for moving targets, *asterisks* indicating difficult-to-search cells, and *circles* indicating initial locations of searchers. The number of searchers is proportional to the radius of a circle

Table 4.1 Scenarios defining problem instances of **SPX**. Columns marked with $j = j'$ ($j \neq j'$) give glimpse-detection probability for a searcher that occupy (not occupy) the current cell previously. An asterisk indicates a column with glimpse-detection probability for difficult-to-search cells; see Fig. 4.2

	Number of searchers		Glimpse-detection probability							
			Searcher class 1				Searcher class 2			
Scenario	S_1	S_2	$j=j'$	$j \neq j'$	$j=j'*$	$j \neq j'*$	$j=j'$	$j \neq j'$	$j=j'*$	$j \neq j'*$
1	2	1	0.50	0.29	0.29	0.16	0.29	0.16	0.16	0.09
2	4	2	0.50	0.29	0.29	0.16	0.29	0.16	0.16	0.09
3	20	10	0.07	0.03	0.03	0.02	0.03	0.02	0.02	0.01

if the searcher occupied the current cell in the last time period ($j = j'$), but the searcher's detection rate is reduced with a factor 0.5 if the searcher just moved into the cell ($j \neq j'$). In view of (4.47), this implies a glimpse-detection probability of 0.29; see Table 4.1. This reduction accounts for the effect, which we have observed in field experiments with actual drones (see Kress and Royset 2008; Royset and Reber 2009), that a searcher often wastes some search time transiting from one cell to another even if the cells are adjacent. Using the model flexibility of **SPX**, we incorporate this effect without resorting to a fine time discretization.

When a searcher occupies one of the cells marked with an asterisk in Fig. 4.1, all detection rates are reduced by a factor of 0.5. These cells represent areas with poor search conditions and consequently low detection rates. This results in a glimpse-detection probability of 0.29 when $j = j'$ and 0.16 when $j \neq j'$. For the class-2 searcher, the detection rate is reduced with a factor of 0.5 compared to class 1 in all situations, with resulting glimpse-detection rates given in Table 4.1.

Scenario 2 is identical to scenario 1 except it has four class-1 searchers and two class-2 searchers. Scenario 3 is identical to scenario 1 except that there are 20 class-1 searchers and 10 class-2 searchers, and the detection rate is reduced with a factor of 0.1 in all situations. The last row of Table 4.1 gives the resulting glimpse-detection probabilities. We note that the total detection rate of the searchers in scenario 3 is identical to that of those in scenario 1. Scenario 3, however, allows more flexibility as the search effort can be spread more widely.

We consider both the situations with and without deconfliction constraints (4.53) and (4.54). In these scenarios, deconfliction amounts to ensuring that at most one searcher occupies a cell each time period and that a searcher is not allowed to move from a cell j to an adjacent cell $j' \in \mathscr{F}^1(j)$ when another searcher makes the opposite move from j' to j. We assume that transit to a distance cell (i.e., a cell $j' \in \mathscr{F}_l^2(j)$) takes place by flying at high altitude, while search is carried out at low altitude. In **SPX**, we incorporate these constraints by setting $n_j = 1$ for all j and $\mathscr{D}(l,j,j',t) = \{(l',j'',j''',t') \mid l' \in \mathscr{L}, j'' = j', j''' = j, t' = t\}$ whenever $j' \in \mathscr{F}^1(j), j' \neq j$ and otherwise $\mathscr{D}(l,j,j',t) = \emptyset$. Since searchers transiting between distance cells can be separated easily by altitude, we allow the routes of such searchers to cross each other as well as to cross over searchers occupying cells.

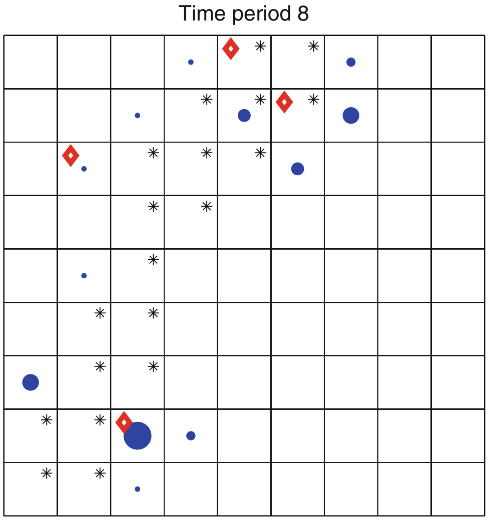


Fig. 4.2 Optimal searcher location during time period 8 for scenario 3, $T = 8$, and no deconfliction constraints. The radius of a circle is proportional to the number of searchers occupying the corresponding cell during time period 8 (cells 4, 7, 12, 14, 16, 20, 24, 38, 55, 66, 67, and 75 contain 1, 2, 1, 3, 4, 1, 3, 1, 4, 7, 2, and 1 searchers, respectively). *Diamonds* indicate initial location for moving targets and *asterisks* indicate difficult-to-search cells

Table 4.2 shows lower and upper bounds on the optimal value of **SPX** as well as the corresponding relative optimality gaps after 15 and 60 min of calculation time of the Cutting Plane Algorithm with $T = 8$, 10, and 12 as obtained with a low-end laptop (of 2010); see Royset and Sato (2010) for details. Columns 4 and 5 present the results for the case with no deconfliction constraints, while columns 6 and 7 include deconfliction constraints. Interestingly, the algorithm solves problem instances with more searchers (scenario 3) quicker than those with fewer searchers (scenario 1) as the linearizations of the nonlinear functions tend to be more accurate in those cases. We also find longer time horizons to be more difficult, primarily due to the weaker cuts in the case of smaller nondetection probabilities.

Deconfliction constraints restrict **SPX** and result in an increase in the optimal value. In scenarios 1 and 2, the change is small due to the relatively low number of searchers. Deconfliction constraints increase the optimal value with about 0.05 in scenario 3 where 30 searchers are present. Hence, deconfliction constraints effectively force the searchers to give up 0.05 in probability of detection.

Figure 4.2 illustrates the optimal location of searchers in time period 8 for scenario 3, $T = 8$, and no deconfliction constraints. The radius of a circle is proportional to the number of searchers located in the corresponding cell during time period 8 (see figure caption). We observe that multiple searchers focus on a relatively small number of cells with high probability of containing targets. Figure 4.3 illustrates the same situation but for the case with deconfliction constraints.

Table 4.3 further examines the upper bounds (UB) on the optimal value of **SPX** and corresponding relative optimality gaps in scenario 3 as T increases. As in Table 4.2, we find a worsening in solution quality as T increases. However, the increase is moderate and essentially insignificant for the case with deconfliction constraints. One is able to obtain near-optimal solutions even for long time horizons.

Table 4.2 Lower and upper bounds on the optimal value of **SPX** as well as relative optimality gaps after 15 and 60 min of calculation times of Cutting Plane Algorithm for scenarios 1–3 with and without deconfliction constraints. The time in seconds to reach optimality is reported in brackets when zero gap is achieved within 60 min

Scenario	T	Measure	No deconfliction After 15 min	No deconfliction After 60 min	Deconfliction After 15 min	Deconfliction After 60 min
1	8	Lower bound	0.851	0.858	0.852	0.858
		Upper bound	0.874	0.866	0.873	0.866
		Relative gap	0.027	0.010	0.024	0.010
2	8	Lower bound	0.719	0.727	0.721	0.731
		Upper bound	0.752	0.752	0.774	0.760
		Relative gap	0.046	0.035	0.072	0.040
3	8	Lower bound	0.837	0.837	0.888	
		Upper bound	0.837	0.837	0.888	
		Relative gap	0.001	0.000	0 [390]	
1	10	Lower bound	0.777	0.796	0.781	0.801
		Upper bound	0.833	0.833	0.842	0.842
		Relative gap	0.072	0.047	0.079	0.052
2	10	Lower bound	0.626	0.635	0.628	0.636
		Upper bound	0.710	0.703	0.711	0.711
		Relative gap	0.133	0.108	0.132	0.117
3	10	Lower bound	0.788	0.788	0.840	0.840
		Upper bound	0.789	0.789	0.841	0.841
		Relative gap	0.002	0.002	0.001	0.000
1	12	Lower bound	0.726	0.742	0.727	0.742
		Upper bound	0.818	0.818	0.815	0.815
		Relative gap	0.127	0.103	0.121	0.099
2	12	Lower bound	0.543	0.554	0.543	0.551
		Upper bound	0.675	0.664	0.654	0.654
		Relative gap	0.243	0.199	0.206	0.189
3	12	Lower bound	0.745	0.745	0.797	0.797
		Upper bound	0.748	0.747	0.799	0.799
		Relative gap	0.004	0.003	0.003	0.002

Interestingly, deconfliction constraints reduce optimality gaps in these instances even though they increase the model size.

In summary, we see that in these rather realistic scenarios involving 30 searchers divided in two classes with different speed and sensor quality, four targets of variable characteristics, deconfliction constraints, 81 inhomogeneous cells, and a long time horizon of up to 42 periods, one can obtain a 6 % near-optimal solution in 60 min on a low-end laptop (of 2010).

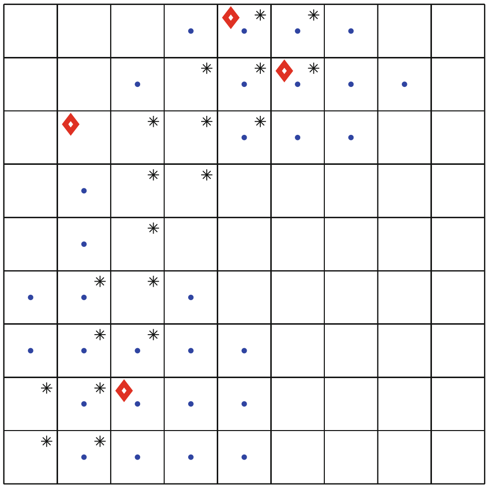


Fig. 4.3 Optimal searcher location during time period 8 for scenario 3, $T = 8$, and deconfliction constraints. A *dot* indicates one searcher

Table 4.3 Upper bounds (*UB*) on the optimal value of **SPX** as well as relative optimality gaps after 60 min of calculation times of Cutting Plane Algorithm for scenario 3 with varying T, and with and without deconfliction constraints

	No deconfliction		Deconfliction	
T	UB	Rel. gap	UB	Rel. gap
10	0.789	0.002	0.841	0.000
14	0.709	0.004	0.760	0.002
18	0.643	0.006	0.691	0.002
22	0.587	0.012	0.629	0.002
26	0.539	0.023	0.576	0.004
30	0.513	0.087	0.527	0.005
34	0.476	0.117	0.483	0.009
38	0.447	0.181	0.445	0.022
42	0.410	0.234	0.411	0.060

4.5 Notes

Stewart (1979, 1980) appear to be the first papers to deal with path-constrained search in discrete time and space. These early studies as well as Eagle and Yee (1990), Martin (1993), Dell et al. (1996), Washburn (1998) and Lau et al. (2008) focus on the development of specialized branch-and-bound algorithms for finding an optimal path for a searcher. Bounds in these algorithms are obtained by replacing the probability of detection with, effectively, the expected number of detections; see Dell et al. (1996), Washburn (1998) and Lau et al. (2008). Alternatively, bounds can be obtained by assuming that the searcher can divide its effort among multiple cells each time period as in Eagle and Yee (1990). The presentation in Sect. 4.1 relies on Sato and Royset (2010).

The formulation **SPX** of the constrained search optimization problem as well as **SP1**, **SP1-L**, and **SP1-LM** are taken from Royset and Sato (2010). A precursor to **SPX** with a single target, no deconflication constraints, and simpler glimpse-detection probabilities is found in Stewart (1979). We reference Pietz and Royset (2013) alternative mathematical programming models for search that avoid discretizing the area of interest into cells.

The cutting-plane algorithm of Sect. 4.3 is a direct application of general methods in the optimization literature; see, e.g., Kelley (1960), Duran and Grossmann (1986), Westerlund and Pettersson (1995) and Bonami et al. (2008). The efficient procedure for computing $f(Y)$ and its gradient in Sect. 4.3.3 is due to Brown (1980).

There are numerous heuristic algorithms for estimating solutions of constrained search problem such as local search and genetic algorithms, see Dell et al. (1996), cross-entropy method, see Sato (2008), myopic optimization with a receding-horizon, see Dell et al. (1996), Grundel (2005), Wong et al. (2005) and Riehl et al. (2007), sequential optimization of each searcher, see Stewart (1979), Wong et al. (2005) and Hollinger and Singh (2008), and decentralized optimization by each searcher, see Bourgult et al. (2003), Yang et al. (2004) and Wong et al. (2005).

References

Ahuja RK, Magnanti TL, Orlin JB (1993) Network flows. Prentice Hall, Upper Saddle River

Bonami P, Biegler L, Conn A, Cornuejols G, Grossmann I, Laird C, Lee J, Lodi A, Margot F, Sawaya N, Wachter A (2008) An algorithmic framework for convex mixed integer nonlinear programs. Discret Optim 5:186–204

Bourgult F, Furukawa T, Durrant-Whyte H (2003) Coordinated decentralized search for a lost target in a bayesian world. In: Proceedings of the 2003 IEEE/RSJ international conference on intelligence and systems, Las Vegas, pp 48–53

Brown S (1980) Optimal search for a moving target in discrete time and space. Oper Res 28(6):1275–1289

Caldwell T (1961) On finding minimal routes in a network with turning penalties. Commun ACM 4:107–108

COIN-OR (2009) Bonmin web page. http://www.coin-or.org/Bonmin/. Last accessed 10 June 2009

Dell R, Eagle J, Martins G, Santos A (1996) Using multiple searchers in constrained-path, moving-target search problems. Nav Res Logist 43:463–480

Duran M, Grossmann I (1986) An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math Program 36:307–339

Eagle J, Yee J (1990) An optimal branch and bound procedure for the constrained path, moving target search problem. Oper Res 38:110–114

Grossmann I, Viswanathan J, Vecchietti A, Raman R, Kalvelagen E (2008) DICOPT user manual. GAMS Development Corporation, Washington, DC

Grundel D (2005) Constrained search for a moving target. In: Proceedings of the 2005 international symposium on collaborative technologies and systems, St. Louis, pp 327–332

Hollinger G, Singh S (2008) Proofs and experiments in scalable, near-optimal search by multiple robots. In: Proceedings of robotics: science and systems conference, Zurich

Kelley J (1960) The cutting plane method for solving convex programs. J Soc Ind Appl Math 8:703–712

Kratzke T, Stone L, Frost J (2010) Search and rescue optimal planning system. In: Proceedings of 13th international conference on information fusion (FUSION 2010), Edinburgh

Kress M, Royset JO (2008) Aerial search optimization model (ASOM) for UAVs in special operations. Mil Oper Res 13(1):23–33

Lau H, Huang S, Dissanayake G (2008) Discounted mean bound for the optimal searcher path problem with non-uniform travel times. Eur J Oper Res 190(2):383–397

Martin GA (1993) A new branch-and-bound procedure for computing optimal search path. Master's thesis, Naval Postgraduate School, Monterey

Pietz J, Royset JO (2013) Generalized orienteering problem with resource dependent rewards. Nav Res Logist 60(4):294–312

Rardin RL (1997) Optimization in operations research. Prentice Hall, Upper Saddle River

Riehl J, Collins G, Hespanha J (2007) Cooperative graph-based model predictive search. In: Proceedings of 46th IEEE conference on decision and control, New Orleans, pp 2998–3004

Royset JO, Reber D (2009) Optimizing routing of unmanned aerial systems for the interdiction of improvised explosive devices. Mil Oper Res 14(4):5–19

Royset JO, Sato H (2010) Route optimization for multiple searchers. Nav Res Logist 57(8): 701–717

Sato H (2008) Path optimization for single and multiple searchers: models and algorithms. PhD thesis, Naval Postgraduate School, Monterey

Sato H, Royset JO (2010) Path optimization for the resource-constrained searcher. Nav Res Logist 57(5):422–440

Stewart T (1979) Search for a moving target when searcher motion is restricted. Comput Oper Res 6(3):129–140

Stewart T (1980) Experience with a branch-and-bound algorithm constrained search motion. In: Haley K, Stone L (eds) Search theory and applications. Plenum Press, New York

Washburn A (1998) Branch and bound methods for a search problem. Nav Res Logist 45:243–257

Westerlund T, Pettersson F (1995) An extended cutting plane method for solving convex MINLP problems. Comput Chem Eng 19:131–136

Wong E, Bourgault F, Furukawa T (2005) Multi-vehicle Bayesian search for multiple lost targets. In: Proceedings of the 2005 IEEE international conference on robotics and automation, Barcelona, pp 3169–3174

Yang Y, Minai A, Polycarpou M (2004) Decentralized cooperative search by networks UAVs in an uncertain environment. In: Proceedings of the 2004 American control conference, Boston, pp 5558–5563

Chapter 5
Search for Moving Targets in Continuous Space

In Chap. 3, we found optimal search allocations for problems that take place in discrete space and time. For the most part, algorithms for computing these allocations are limited to exponential detection functions.

This chapter develops methods for finding optimal plans for search problems in discrete or continuous time that take place in continuous space. Sections 5.1, 5.2 and 5.3 consider discrete time searches. Section 5.2 finds optimal detection search plans for exponential detection functions. Section 5.3 finds optimal search plans for decreasing-rate detection functions. Section 5.4 considers a more general class of payoff functions with search-effort like constraints. The effort allocations take place in continuous or discrete time over a continuous space. This class of payoff functions encompasses detection functions as well as the FAB payoff functions defined in Chap. 3. We find necessary and sufficient conditions for optimal plans for these payoff functions.

5.1 Search Problem: Discrete Time

As in previous chapters, the search takes place in discrete time over the interval $[0, T]$. The times are represented by the integers $t = 0, \ldots, T$; the increments between times t and $t + 1$ need not be equal for all t. For convenience we will use the notation $[0, T]$ to stand for the "interval" of integers from 0 to T when time is discrete. The target search space S is continuous and is usually a subset of 2 or 3-dimensional space.

Electronic supplementary material The online version of this chapter (doi: 10.1007/978-3-319-26899-6_5) contains supplementary material, which is available to authorized users.

L.D. Stone et al., *Optimal Search for Moving Targets*, International Series in Operations Research & Management Science 237,
DOI 10.1007/978-3-319-26899-6_5

Prior Distribution on Target Motion The prior distribution on target motion is specified by a stochastic process $X = \{X(t),\ t = 0, \ldots, T\}$ where $X(t) \in S$ for $t = 0, \ldots, T$. Let $\omega = (\omega_0, \ldots, \omega_T)$ denote a sample path of the process so that ω_t is the target's state at time t. Let Ω be the set of sample paths and p be the probability density function on Ω.

Search Plans A search plan is a space-time allocation f defined on $S \times [0, T]$where $f(x, t)$ is the effort density applied to x at time t. As in Chap. 3 we suppose that $m(t)$ search effort is available at time t and define the class $F(m)$ of continuous-space, discrete-time search plans as follows.

$$
\begin{aligned}
&f \in F(m) \text{ if and only if} \\
&\quad 0 \leq f(x,t) < \infty \ x \in S \text{ and } t = 0, \ldots, T \\
&\quad \int_S f(x,t)\, dx = m(t) \text{ for } t = 0, \ldots, T.
\end{aligned}
\tag{5.1}
$$

There may be an upper bound B on the search density. In this case we define the class $F_B(m)$ of search plans where

$$
\begin{aligned}
&f \in F_B(m) \text{ if and only if} \\
&\quad f \in F(m) \\
&\quad f(x,t) \leq B \text{ for } x \in S \text{ and } t = 0, \ldots, T.
\end{aligned}
\tag{5.2}
$$

If $B = \infty$, then $F_B(m) = F(m)$.

Detection Function As in Chap. 3, the probability of detecting the target, given it follows path ω, is a function of the weighted total search effort density that "falls on" the target as it follows that path. The function

$$
\zeta(f, \omega, t) = \sum\nolimits_{s=0}^{t} W(\omega_s, s) f(\omega_s, s) \quad \text{for } \ \omega \in \Omega,\ t = 0, \ldots, T \tag{5.3}
$$

accumulates the weighted search effort density over $[0, t]$ for the path ω where the weight $W(x, s)$ represents the relative detectability or sweep width for the target given it is located at x at time s. There is a detection function b such that

$$
b(\zeta(f, \omega, t)) = \Pr\{\text{detecting the target by time } t | \text{target follows path } \omega\}. \tag{5.4}
$$

Probability of Detection The probability of detection by time t is

$$
P(f, t) = E[b(\zeta(f, \omega, t))] \quad \text{for } \ t = 0, \ldots, T \tag{5.5}
$$

where $E[\cdot]$ indicates expectation over the probability distribution p on the sample paths of X.

***T*-Optimal Plan** A plan $f^* \in F_B(m)$ is *T-optimal* if and only if

$$P\left(f^*, T\right) \geq P\left(f, T\right) \ \text{for} f \in F_B(m). \tag{5.6}$$

In most cases, we cannot construct a moving target plan that is uniformly optimal. This means that the plan that is optimal for $T + \Delta T$ is not an extension of the T-optimal plan. One has to choose the time at which he wishes the plan to be optimal. This makes it more difficult to find plans that minimize mean time to detection.

5.1.1 Bound on Optimal Plan: Decreasing-Rate Detection Function

One can easily adapt the argument in Sect. 3.1.2 to find an upper bound on the probability of detection for the T-optimal plan when the search space is continuous. As in Chap. 3, this bound will be useful in providing a stopping criterion for algorithms that find approximations to optimal plans.

Suppose b is a decreasing-rate detection function. Let E_{xt} indicate expectation conditioned on $X(t) = x$, and $p_t(x)$ be the probability density for $X(t) = x$. Let $f_1, f_2 \in F_B(m)$ be two allocation functions. Because b is a decreasing-rate detection function, we have

$$\begin{aligned} P\left(f_2, T\right) - P\left(f_1, T\right) &= E\left[b\left(\zeta\left(f_2, \omega, T\right)\right)\right] - E\left[b\left(\zeta\left(f_1, \omega, T\right)\right)\right] \\ &\leq E\left[b'\left(\zeta\left(f_1, \omega, T\right)\right)\left(\zeta\left(f_2, \omega, T\right) - \zeta\left(f_1, \omega, T\right)\right)\right]. \end{aligned} \tag{5.7}$$

Define

$$D\left(f, x, t\right) = E_{xt}\left[b'\left(\zeta\left(f, \omega, T\right)\right)\right] p_t(x) W\left(x, t\right) \ \text{for} f \in F(m). \tag{5.8}$$

Following the proof in Sect. 3.1.2 and replacing summation over $j = 1, \ldots, J$ with integration over S we obtain

$$\begin{aligned} &E\left[b'\left(\zeta\left(f_1, \omega, T\right)\right)\left(\zeta\left(f_2, \omega, T\right) - \zeta\left(f_1, \omega, T\right)\right)\right] \\ &\qquad = \sum_{t=0}^{T} \int_S D\left(f_1, x, t\right)\left(f_2\left(x, t\right) - f_1\left(x, t\right)\right) dx \end{aligned} \tag{5.9}$$

so that by (5.7)

$$P\left(f_2, T\right) - P\left(f_1, T\right) \leq \sum_{t=0}^{T} \int_S D\left(f_1, x, t\right)\left(f_2\left(x, t\right) - f_1\left(x, t\right)\right) dx. \tag{5.10}$$

The inequality in (5.7) holds for any concave detection function b provided that b' is understood to be the subgradient of b.

Let

$$\overline{\lambda}(t) = \sup_{x \in S} D\left(f_1, x, t\right) \text{ and } \underline{\lambda}(t) = \inf_{\{x: f_1(x,t) > 0\}} D\left(f_1, x, t\right) \quad \text{for } t = 0, \ldots, T. \tag{5.11}$$

In the case, where $f_1(x, t) = 0$ for $x \in S$, we set $\underline{\lambda}(t) = 0$. From (5.10) we have

$$\begin{aligned} P\left(f_2, T\right) - P\left(f_1, T\right) &\leq \sum\nolimits_{t=0}^{T} \int_S \left[\overline{\lambda}(t) f_2(x, t) - \underline{\lambda}(t) f_1(x, t)\right] dx \\ &= \sum\nolimits_{t=0}^{T} \left(\overline{\lambda}(t) - \underline{\lambda}(t)\right) m(t) \equiv \Delta\left(f_1\right). \end{aligned} \tag{5.12}$$

Notice that the right-hand side of the second line of (5.12) does not depend on f_2. Thus if $f^* \in F_B(m)$ is T-optimal, then

$$P\left(f^*, T\right) \leq P(f, T) + \Delta(f) \text{ for any } f \in F_B(m).$$

We can now state the upper bound theorem obtained by Washburn (1981).

Theorem 5.1. *If b is an decreasing-rate detection function and $f^* \in F_B(m)$ is T-optimal, then*

$$P\left(f^*, T\right) \leq P(f, T) + \Delta(f) \text{ for any } f \in F_B(m) \tag{5.13}$$

where

$$\Delta(f) = \sum\nolimits_{t=0}^{T} \left(\overline{\lambda}(t) - \underline{\lambda}(t)\right) m(t).$$

Measure Theory Considerations Those readers familiar with measure theory will realize that we have to interpret sup and inf as ess sup and ess inf in (5.11).

5.2 *T*-Optimal Plan: Exponential Detection Function, Discrete Time

In this section we find necessary and sufficient conditions for a plan to be T-optimal when the detection function is exponential and time is discrete. Using these conditions, we extend Brown's recursion in Sect. 3.2.1 for finding a T-optimal plan in discrete space and time to continuous space and discrete time. We finish with a discussion of a method for implementing this recursion in an approximate fashion.

5.2.1 Necessary and Sufficient Conditions for a T-Optimal Plan: Exponential Detection Function

As in Chap. 3, we show that when the detection function is exponential, finding a T-optimal plan is equivalent to solving a sequence of stationary target problems. The proof will be similar to the one given in Sect. 3.2, but since the search space is continuous rather than discrete, we cannot rely on the Kuhn-Tucker-Karush theorem to provide necessary and sufficient conditions for optimality.

Theorem 5.1. *Assume b is an exponential detection function. Then a necessary and sufficient condition for $f^* \in F_B(m)$ to be a T-optimal plan is that for $t = 0, \ldots, T$, $f^*(\cdot, t)$ is the optimal stationary target plan for cost $m(t)$ for the distribution $\tilde{q}(\cdot, t, f^*)$, which is the posterior probability distribution on the target's location at time t given failure to detect at all times other than t.*

Proof. Since $b(z) = 1 - e^{-z}$, we have from (5.3) and (5.5) that for any $t = 0, \ldots, T$,

$$\begin{aligned} 1 - P(f, T) &= E\left[\exp\left(-\zeta\left(f, \omega, T\right)\right)\right] = E\left[\exp\left(-\sum\nolimits_{s=0}^{T} W\left(\omega_s, s\right) f\left(\omega_s, s\right)\right)\right] \\ &= \int_S E_{xt}\left[\exp\left(-\sum\nolimits_{s=0}^{T} W\left(\omega_s, s\right) f\left(\omega_s, s\right)\right)\right] p_t(x) dx \\ &= \int_S e^{-W(x,t) f(x,t)} E_{xt}\left[\exp\left(-\sum_{s \neq t} W\left(\omega_s, s\right) f\left(\omega_s, s\right)\right)\right] p_t(x) dx \\ &= \int_S e^{-W(x,t) f(x,t)} q\left(x, t, f\right) dx \end{aligned} \tag{5.14}$$

where

$$q\left(x, t, f\right) = E_{xt}\left[\exp\left(-\sum_{s \neq t} W\left(\omega_s, s\right) f\left(\omega_s, s\right)\right)\right] p_t(x). \tag{5.15}$$

One can see that $q(x, t, f)$ is the probability density that the target is in state x at time t and not detected by the search at any time other than t. In addition, from (5.8) and (5.15) we obtain

$$
\begin{aligned}
D(f,x,t) &= E_{xt}\left[b'(\zeta(f,\omega,T))\right]p_t(x)W(x,t) \\
&= E_{xt}\left[\exp\left(-\sum\nolimits_{s=0}^{T} W(\omega_s,s)f(\omega_s,s)\right)\right]p_t(x)W(x,t) \\
&= e^{-W(x,t)f(x,t)}E_{xt}\left[\exp\left(-\sum_{s\neq t} W(\omega_s,s)f(\omega_s,s)\right)\right]p_t(x)W(x,t) \\
&= W(x,t)\,e^{-W(x,t)f(x,t)}q(x,t,f) \ \text{ for } f \in F_B.
\end{aligned}
\tag{5.16}
$$

Necessity Suppose $f^* \in F_B(m)$ is a T-optimal plan. Since $q(\cdot,t,f^*)$ is proportional to $\tilde{q}(\cdot,t,f^*)$, it is clear from (5.14) that $f^*(\cdot,t)$ must minimize the failure probability (i.e., maximize detection probability) for the stationary target problem with distribution $q(\cdot,t,f^*)$ among plans with cost $m(t)$ for $t = 0,\ldots,T$. This shows that maximizing detection probability for the stationary target problem defined by $\tilde{q}(\cdot,t,f^*)$ for $t = 0,\ldots,T$ is a necessary condition for T-optimality.

Sufficiency Suppose that $f^*(\cdot,t)$ maximizes the detection probability for the stationary target search corresponding to $\tilde{q}(\cdot,t,f^*)$ (and therefore $q(\cdot,t,f^*)$) among plans in F_B with effort $m(t)$ for $t = 0,\ldots,T$.

Recall from Sect. 2.3.3.1 that the pointwise Lagrangian l is defined by

$$l(x,z,\lambda) = b(x,z)p(x) - \lambda c(x)z \ \text{ for } \ x \in S,\ 0 \le z \le B, \ \text{and } \lambda > 0$$

and that its derivative with respect to z is defined by

$$l'(x,z,\lambda) = b'(x,z)p(x) - \lambda c(x).$$

Since $c(x) = 1$ and $b(x,z) = 1 - e^{-W(x,t)z}$ for $t = 0,\ldots,T$, the derivative becomes

$$l'(x,z,\lambda) = W(x,t)\,e^{-W(x,t)z}p(x) - \lambda. \tag{5.17}$$

The remark made after Theorem 2.3 notes that conditions (2.65) are necessary and sufficient for a stationary target plan $f^*(\cdot,t)$ to be optimal for its cost $m(t)$. For the problem considered here, these conditions become

$$
\begin{aligned}
W(x,t)\,e^{-W(x,t)f^*(x,t)}q\left(x,t,f^*\right) &\ge \lambda_t \ \text{ for } f^*(x,t) = B \\
&= \lambda_t \ \text{ for } 0 < f^*(x,t) < B \\
&\le \lambda_t \ \text{ for } f^*(x,t) = 0.
\end{aligned}
\tag{5.18}
$$

By (5.16) we may write (5.18) as

$$
\begin{aligned}
D\left(f^*, x, t\right) &\geq \lambda_t \text{ for } f^*(x, t) = B \\
&= \lambda_t \text{ for } 0 < f^*(x, t) < B \\
&\leq \lambda_t \text{ for } f^*(x, t) = 0.
\end{aligned} \tag{5.19}
$$

Since $f^*(\cdot, t)$ is T-optimal, it must satisfy (5.19) for $t = 0, \ldots, T$. By (5.10) and (5.19), we have

$$
\begin{aligned}
P(f, T) - P(f^*, T) &\leq \sum\nolimits_{t=0}^{T} \int_S D\left(f^*, x, t\right)\left(f(x, t) - f^*(x, t)\right) dx \\
&\leq \sum\nolimits_{t=0}^{T} \lambda_t \int_S \left(f(x, t) - f^*(x, t)\right) dx = 0.
\end{aligned} \tag{5.20}
$$

where the inequality in the last line of (5.20) follows by breaking the integral over S into the sum of integrals over S_1, S_2, and S_3 where

$$S_1 = \{x : f^*(x, t) = B\}, \; S_2 = \{x : 0 < f^*(x, t) < B\}, \; S_3 = \{x : f^*(x, t) = 0\}.$$

On the set S_1, $D(f^*, x, t) \geq \lambda_t > 0$ and $f(x, t) \leq f^*(x, t) = B$, so

$$\int_{S_1} D\left(f^*, x, t\right)\left(f(x, t) - f^*(x, t)\right) dx \leq \int_{S_1} \lambda_t \left(f(x, t) - f^*(x, t)\right) dx.$$

On the set S_2, $D(f^*, x, t) = \lambda_t$, so

$$\int_{S_2} D\left(f^*, x, t\right)\left(f(x, t) - f^*(x, t)\right) dx = \int_{S_2} \lambda_t \left(f(x, t) - f^*(x, t)\right) dx,$$

and on the set S_3, $0 \leq D(f^*, x, t) \leq \lambda_t$, and $f(x, t) \geq f^*(x, t) = 0$, so

$$\int_{S_3} D\left(f^*, x, t\right)\left(f(x, t) - f^*(x, t)\right) dx \leq \int_{S_3} \lambda_t \left(f(x, t) - f^*(x, t)\right) dx.$$

The sum of these three integrals yields the inequality in the last line of (5.20). The equality in this line follows from the fact that $f, f^* \in F_B(m)$. Thus

$$P(f, T) \leq P\left(f^*, T\right) \quad \text{for } f \in F_B(m)$$

which proves sufficiency and completes the proof of Theorem 5.1.

5.2.2 *Recursion for Finding T-Optimal Plan for an Exponential Detection Function*

The following version of Brown's recursion is identical to the one in Sect. 3.2.1 except that the target state space is continuous rather than discrete.

T-Optimal Search Plan Recursion for a Continuous-Effort, Exponential Detection Function

1. Let $f_0(x,t) = 0$ for $x \in S$ and $t = 0, \ldots, T$.
2. Let $\varepsilon > 0$ be a tolerance.
3. Set $k = 0$.
4. Set $s = k\,[\mathrm{mod}\,(T+1)]$, i.e., s is the integer remainder after dividing k by $T+1$.
5. Compute $\xi(\cdot, s, f_k)$, the optimal plan for $q(\cdot, s, f_k)$ with $m(s)$ effort.
6. Set

$$f_{k+1}(\cdot, t) = \begin{cases} f_k(\cdot, t) & \text{for } t \neq s \\ \xi(\cdot, s, f_k) & \text{for } t = s. \end{cases}$$

7. If $s = T$, compute $\Delta(f_{k+1})$. If this is less than ε stop; $P(f_{k+1})$ is within ε of optimal.
8. Otherwise set $k = k + 1$, and go back to step 4.

Since the recursion starts with $f_0(\cdot, t) = 0$ for $t = 0, \ldots, T$, the first pass through the recursion from $k = 0$ to T produces the myopic search plan, the one that maximizes the increase in detection probability at each time increment. Since each allocation f_{k+1} increases the detection probability compared to f_k, for $k = 0, 1, \ldots,$ $P(f_k, T)$ approaches a limit $\overline{P}$ as $k \to \infty$. Thus we can obtain a plan as close to optimal as we please. If we reach a step k in the recursion where $f_{k+T+1} = f_k$, then f_k satisfies the necessary and sufficient conditions for optimality and is an optimal plan.

Implementing the Recursion We can implement the above recursion in an approximate fashion by replacing the target motion process by a discrete number of sample paths. To do this, we draw a large number N of sample paths from the target motion process. This is usually done by a simulation that produces sample paths ω each with $p(\omega) = 1/N$. These sample paths develop in continuous space and time. We impose a grid of J cells on the search space and make the further approximation that we will restrict ourselves to plans that place effort uniformly within any given cell. In effect we can allocate effort to a cell but not to a portion of a cell. For each simulated path ω, we compute $\widehat{\omega}_t$, the cell containing the target at time t, for

$t = 0, \ldots, T$. Replace ω by its discretized sampled path $\widehat{\omega} = (\widehat{\omega}_0, \widehat{\omega}_1, \ldots)$. We then use these N discretized sample to paths produce the distribution of the target's location in the grid for each time t. Note, the target motion is still in continuous space and time; we are simply obtaining a cellular approximation to the target distribution at the search times. One can now use the algorithm given in Sect. 3.2.2 to compute an approximation to the T-optimal plan. See Chap. 5 in Shapiro et al. (2009) for a discussion of the statistical properties of solutions obtained based on Monte Carlo sampling of target paths.

5.3 T-Optimal Plan: Decreasing-Rate Detection Function, Discrete Time

In this section we extend the methods used to find T-optimal plans for exponential detection functions to the class of decreasing-rate detections functions. As before, time is discrete.

5.3.1 *Necessary and Sufficient Conditions for a T-Optimal Plan: Decreasing-Rate Detection Function*

The necessary and sufficient conditions given in (5.26) below are stated in terms of the following stationary target problem. From (5.3) and (5.5), we have

$$\begin{aligned} P(f, T) &= E\left[b\left(\zeta\left(f, \omega, T\right)\right)\right] \\ &= E\left[b\left(W(\omega_t, t) f(\omega_t, t) + \sum_{s \neq t} W(\omega_s, s) f(\omega_s, s)\right)\right] \\ &= \int_S E_{xt}\left[b\left(W(x, t) f(x, t) + \sum_{s \neq t} W(\omega_s, s) f(\omega_s, s)\right)\right] p_t(x) dx \text{ for } t = 0, \ldots, T. \end{aligned} \tag{5.21}$$

Define

$$\beta_f(x, t, z) = E_{xt}\left[b\left(W(x, t)\, z + \sum_{s \neq t} W(\omega_s, s) f(\omega_s, s)\right)\right] \text{ for } x \in S \text{ and } z \geq 0. \tag{5.22}$$

Since $\beta_f(x,t,\cdot)$ is the sum of concave functions, it is concave. Since $b'(0) < \infty$ and $b'(u)$ is a decreasing function of u, we have $(b(u+h) - b(u))/h \leq b'(0)$ for all $u, h > 0$. Thus we can use the dominated convergence theorem to compute

$$\beta_f'(x,t,z) = E_{xt}\left[b'\left(W(x,t)z + \sum_{s\neq t} W(\omega_s, s) f(\omega_s, s)\right)\right] W(x,t) \quad \text{for } x \in S. \tag{5.23}$$

Observe that $\beta_f(x,t,\cdot)$ is a decreasing-rate detection function. Let G_B be the set of stationary target search plans $g : S \to [0,B]$. If $B = \infty$,then $g : S \to [0,\infty)$. Define

$$\widehat{P}_f(t,g) = \int_S \beta_f(x,t,g(x))\, p_t(x)dx \text{ for } g \in G_B \tag{5.24}$$

Finding $g \in G_B$ to maximize $\widehat{P}_f(t,g)$, subject to a constraint on effort, is a stationary target problem with probability distribution $p_t(x)$ and detection function $\beta_f(x,t,\cdot)$ for $x \in S$. From (5.21) and (5.22), we have

$$P(f,T) = \widehat{P}_f(t, f(\cdot,t)) \quad \text{for } t = 0,\ldots,T \text{ and } f \in F_B(m). \tag{5.25}$$

Theorem 5.2. *If b is a decreasing-rate detection function, then both (5.26) and (5.27) below are necessary and sufficient conditions for $f^* \in F_B(m)$ to be T-optimal.*

$$\widehat{P}_f(t, f^*(\cdot,t)) \geq \widehat{P}_f(t,g) \text{ for } g \in G_B \text{ such that } \int_S g(x)dx = m(t) \quad \text{for } t = 0,\ldots,T. \tag{5.26}$$

For $t = 0,\ldots,T$, there exists a $\lambda_t \geq 0$, such that

$$\begin{aligned} D(f^*, x, t) &\geq \lambda_t \text{ for } f^*(x,t) = B \\ &= \lambda_t \text{ for } 0 < f^*(x,t) < B \\ &\leq \lambda_t \text{ for } f^*(x,t) = 0 \qquad \text{for } x \in S. \end{aligned} \tag{5.27}$$

Proof. We first prove necessity.

Necessity Suppose $f^* \in F_B(m)$ *is* T-optimal. Statement (5.26) follows by contradiction. Suppose that for some t, there is a $g \in G_B$ such that

$$\int_S g(x)dx = m(t) \text{ and } \widehat{P}_f(t,g) > \widehat{P}_f\left(t, f^*(\cdot,t)\right).$$

Then we could define

$$f(\cdot, s) = \begin{cases} g & \text{for } s = t \\ f^*(\cdot, s) & \text{for } s \neq t \end{cases}$$

such that $f \in F_B(m)$ and $P(f, T) > P(f^*, T)$ which contradicts the T-optimality of f^*. Thus (5.26) holds for $t = 0, \ldots, T$.

Since (5.26) holds, we have that $f^*(\cdot, t)$ is an optimal stationary target plan for $m(t)$ effort for the stationary target distribution given by $p_t(x)$ and decreasing rate detection function $\beta_f(x, t, \cdot)$ for $x \in S$. By the remark after Corollary 2.3, conditions (2.65) are necessary for the optimality a stationary target plan in G_B. In terms of this stationary target problem, these conditions become

$$\begin{aligned} \beta'_{f^*}(x, t, f^*(x, t))\, p_t(x) &\geq \lambda_t \text{ for } f^*(x, t) = B \\ &= \lambda_t \text{ for } 0 < f^*(x, t) < B \\ &\leq \lambda_t \text{ for } f^*(x, t) = 0 \qquad \text{for } x \in S. \end{aligned} \tag{5.28}$$

By (5.8) and (5.23), $\beta'_{f^*}(x, t, f^*(x, t))\, p_t(x) = D(f^*, x, t)$ and (5.27) follows.

Sufficiency Suppose conditions (5.27) hold. Note that these are the same conditions as in (5.19). With this in mind, we can use the portion of the proof of Theorem 5.1 that starts at the second sentence after (5.19) to prove that f^* is T-optimal.

Suppose conditions (5.26) hold. Then as we have shown above, conditions (5.27) must hold and f^* is T-optimal. This proves the theorem.

5.3.2 *Recursion for T-Optimal Plan for a Decreasing-Rate Detection Function*

The recursion for finding a T-optimal plan for a decreasing-rate detection function requires solving a solving a sequence of stationary target problems that involve maximizing $\widehat{P}_f(t, g)$ defined in (5.24) over allocations g with effort $m(t)$. For these problems, the stationary target distribution is $p_t(x)$ and detection function is $\beta_f(x, t, \cdot)$ for $x \in S$ as given in (5.22).

For $f \in F_B(m)$, let $\xi(\cdot, t, f)$ be the allocation in G_B such that

$$\int_S \xi(x, t, f)\, dx = m(t)$$

and

$$\widehat{P}_f(t, \xi(\cdot, t, f)) \geq \widehat{P}_f(t, g) \text{ for } g \in G_B \text{ such that } \int_S g(x) dx = m(t). \tag{5.29}$$

Since $\beta_f(x, t, \cdot)$ is a decreasing-rate detection function, $\xi(\cdot, t, f)$ can be found by the algorithm in Sect. 2.3.3.2.

The recursion given below is the same as the one in Sect. 5.2.2 with exception of step 5 where the plan obtained is the optimal plan for the payoff function $\widehat{P}_f(s,\cdot)$ rather than the plan that maximizes the probability of detecting the target for the stationary target distribution that is the posterior probability density on target location at time s given failure to detect at all times other than s.

As in Sect. 5.2.2, the first pass through times $t = 0,\ldots,T$ in the recursion produces the myopic plan. Also $P(f_{k+1},T) \geq P(f_k,T)$ for $k = 0,1,\ldots$. This follows from the definition of $f_{k+1}(\cdot,t)$ and (5.25). By the proof given in Sect. 5.2.2, $P(f_k,T)$ approaches a limit $\overline{P}$ as $k \to \infty$, and this limit is P^*, the detection probability for the T-optimal plan.

T-Optimal Search Plan Recursion for a Decreasing-Rate Detection Function

1. Let $f_0(x,t) = 0$ for $x \in S$ and $t = 0,\ldots,T$.
2. Let $\varepsilon > 0$ be a tolerance.
3. Set $k = 0$.
4. Set $s = k\,[\bmod\,(T+1)]$, i.e., s is the integer remainder after dividing k by $T+1$.
5. Compute $\xi(\cdot,s,f_k)$, the allocation that maximizes $\widehat{P}_f(s,\cdot)$ in (5.29).
6. Set

$$f_{k+1}(\cdot,t) = \begin{cases} f_k(\cdot,t) & \text{for } t \neq s \\ \xi(\cdot,s,f_k) & \text{for } t = s. \end{cases}$$

7. If $s = T$, compute $\Delta(f_{k+1})$. If this is less than ε stop; $P(f_{k+1})$ is within ε of optimal.
8. Otherwise set $k = k+1$, and go back to step 4.

Implementing the Recursion As in Sect. 5.2.2, we can implement the above recursion by approximating the target motion process. To do this we draw a large number N of sample paths from the target motion process. This is usually done by a simulation that produces sample paths ω each with $p(\omega) = 1/N$. These sample paths develop in continuous space and discrete time. We impose a grid of J cells on the search space. For each simulated path ω, we compute $\widehat{\omega}_t$, the cell containing the target at time t,for $t = 0,\ldots,T$. Replace ω by its discretized sampled path $\widehat{\omega} = (\widehat{\omega}_0,\widehat{\omega}_1,\ldots)$. Let $\widehat{\Omega}$ be the set of these N discretized sample paths. There are two approximations involved in using $\widehat{\Omega}$. The first is approximating the continuum of paths in Ω with a finite discrete set $\widehat{\Omega}$. The second is in restricting ourselves to allocations that have a constant value over the cells in the grid chosen. The choice of the grid does not affect the motion of the paths in $\widehat{\Omega}$. We are simply recording which grid cell they fall into at each time. The quality of this approximation will depend on the number of sample paths N and the size of the grid cells. The algorithm below finds a T-optimal plan for this approximation.

Implementation of T-Optimal Search Plan Recursion for a Decreasing-Rate Detection Function

1. Set $\zeta(\widehat{\omega}) = 0$ for $\widehat{\omega} \in \widehat{\Omega}$ and $f_0(j,t) = 0$ for $j = 1,\ldots,J$ and $t = 0,\ldots,T$.
2. Let $\varepsilon > 0$ be a tolerance, and set $k = 0$.
3. Set $s = k\,[\mathrm{mod}\,(T+1)]$.
4. Compute

$$\zeta_s(\widehat{\omega}) = \zeta(\widehat{\omega}) - W(\widehat{\omega}_s, s) f_k(\widehat{\omega}_s, s) \quad \text{for } \widehat{\omega} \in \widehat{\Omega}, \tag{5.30}$$

and for $j = 1,\ldots,J$, compute

$$r'(j,s,z) = W(j,s) \sum_{\{\widehat{\omega}:\widehat{\omega}_s = j\}} \Pr\{\widehat{\omega}\}\, b'(W(j,s)\,z + \zeta_s(\widehat{\omega})) \quad \text{for } z \geq 0. \tag{5.31}$$

5. Find $\xi_k(\cdot,s)$, the plan with $m(s)$ effort that maximizes $\widehat{P}_{f_k}(s,\cdot)$, by finding λ_s and $\xi_k(\cdot,s)$, such that

$$r'(j,s,\xi_k(j,s)) \quad \begin{array}{l} \geq \lambda_t \text{ for } \xi_k(j,s) = B \\ = \lambda_t \text{ for } 0 < \xi_k(j,s) < B \\ \leq \lambda_t \text{ for } \xi_k(j,s) = 0 \qquad \text{for } j = 1,\ldots,J \end{array} \tag{5.32}$$

and

$$\sum_{j=1}^{J} \xi_k(j,s) = m(s).$$

6. Set

$$f_{k+1}(\cdot,t) = \begin{cases} f_k(\cdot,t) & \text{for } t \neq s \\ \xi_k(\cdot,s) & \text{for } t = s. \end{cases}$$

7. Set $\zeta(\widehat{\omega}) = \zeta_s(\widehat{\omega}) + W(\widehat{\omega}_s, s) f_{k+1}(\widehat{\omega}_s, s)$ for $\widehat{\omega} \in \widehat{\Omega}$.
8. If $s = T$, compute $\Delta(f_{k+1})$. If it is less than ε, stop. Otherwise set $k = k + 1$, and return to step 3.

Step 5, may be accomplished by a numerical search on λ_t as follows. For each value of λ_t considered, find $g : \{1,\ldots,J\} \to [0,B]$ that satisfies (5.32) and compute

$$m_g(s) = \sum_{j=1}^{J} g(j).$$

If $m_g(s) > m(s)$, increase the value of λ_t. If it is less, decrease it. Continue until $m_g(s) = m(s)$. To calculate $\Delta(f_{k+1})$ in step 8, we use (5.8) to compute

$$D(f_{k+1}, j, t) = W(j, t) \sum_{\{\widehat{\omega}:\widehat{\omega}_t = j\}} \Pr\{\widehat{\omega}\} b'(\zeta(\widehat{\omega})) \quad \text{for } j = 1, \ldots, J, \ t = 0, \ldots, T. \tag{5.33}$$

5.3.3 *Example: Inverse Cube Detection Function*

In this section, we return to the example in Sect. 3.2.3 where we compared the T-optimal plan to the myopic one for a search for a boat. In Sect. 3.2.3, we assume that the detection function is exponential. For this example, we will assume an inverse cube detection function.

In this example, the target, possibly a drug smuggling boat, is known to have left port at the point $(0, 0)$ at time $t = 0$ hrs. There are two possible scenarios for the motion of the target, each of which has equal weight. In scenario 1, the boat travels toward a port at $(0, 480)$ moving at approximately 20 kn. In scenario 2, the boat heads east-northeast at about 20 kn.

Figure 3.1 of Chap. 3, which is reproduced below as Fig. 5.1, shows the target distribution resulting from the two scenarios at the times 6, 12, and 18 h. The part of the distribution corresponding to scenario 1 shows the target heading north to the port at $(0, 480)$. In this scenario, the distribution starts at $(0, 0)$ and heads north while spreading out in the east–west direction until 12 h at which time it starts to condense and eventually ends up at $(0, 480)$ at 24 h. The part of the distribution corresponding to scenario 2 spreads out and moves in an east-northeasterly direction. To compute these distributions, we simulated 50,000 equally weighted target paths in continuous space and time, 25,000 for each scenario. At the search times we imposed a grid of cells 20 nm by 20 nm on a side. This grid of cells represents the grid on which we can allocate search effort. That is we are restricted to plans whose allocation of effort to a cell in this grid must be uniform over the points in that cell. This is a

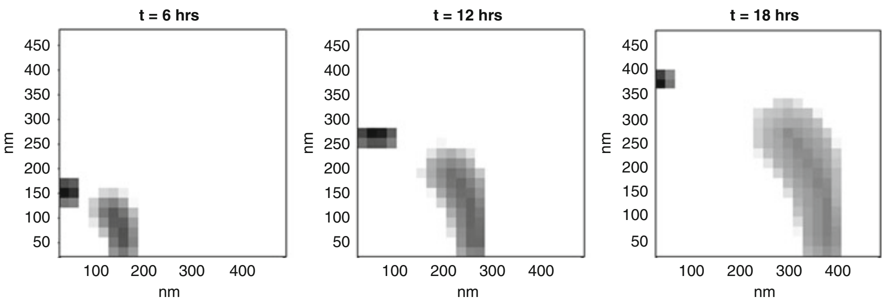


Fig. 5.1 Target location distributions–*dark cells* indicate high probability

reasonable restriction for most searches. Thus the class of allocations over which we optimize are those that allocate effort to grid cells but not to portions of a cell. To accommodate this restriction, we replaced the position of each particle with the index of the cell it is in at the search times. This produces the distribution of the target's location in these cells at the time of the searches. Note, we have not changed the continuous time and space character of the target motion model. We have simply limited the class of allowable search plans.

As in Sect. 3.2.3, there are 3000 nm^2 of search effort (swept area) available at each of the times 6, 12, and 18 h, and the detection function is inverse cube with

$$b(z) = 2\Phi\left(\sqrt{\frac{\pi}{2}}z\right) - 1 = \sqrt{\frac{2}{\pi}}\int_0^{\sqrt{\frac{\pi}{2}}z} e^{-y^2/2}dy \text{ and } b'(z) = e^{-\pi z^2/4} \text{ for } z \geq 0 \tag{5.34}$$

for all times and cells where z is the search effort density in a cell, i.e., the swept area of search effort allocated to a cell divided by 400 nm^2, the area of a cell. As in Chap. 3, we assume that effort can be distributed instantaneously over space at the search times.

We used the algorithm for a decreasing-rate detection function given above to calculate the T-optimal plan for 18 h. Figures 5.2 and 5.3 show the myopic and T-optimal search plans for $T = 18$ hrs at the three search times. To compute these plans, we set $\varepsilon = 0.01$, and in 8 iterations the Washburn bound fell below this value. This produced a detection probability 0.77 for the myopic plan and 0.86 for the T-optimal plan. Looking at Fig. 5.2, we see that the myopic plan allocates substantial effort to scenario 1 at times 6 and 12 h and doesn't look ahead to see that the scenario 2 distribution is spreading out. In Fig. 5.3, the T-optimal plan takes this into account and applies most of its effort to scenario 2 at times 6 and 12 h while waiting until 18 h to apply substantial effort to scenario 1.

The detection probabilities that are obtained in Sect. 3.2.4 for the myopic and T-optimal plans for the exponential detection function are 0.68 and 0.76, about 0.1 lower than with the inverse cube detection function. Looking at Fig. 2.4 which shows a plot of these two detection functions, we see that the inverse cube function lies

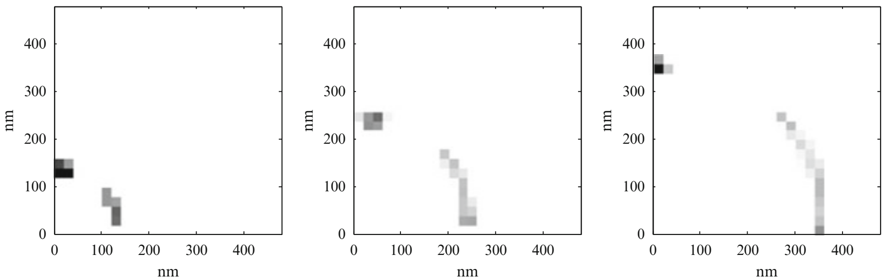


Fig. 5.2 Myopic plan at 6, 12, and 18 h

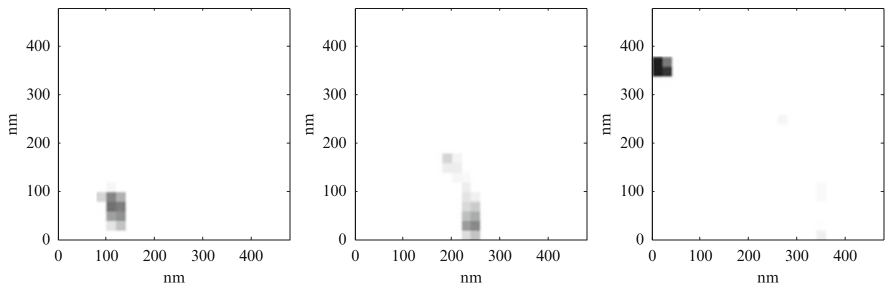


Fig. 5.3 T-optimal plan at 6, 12, and 18 h

above the exponential. It is a more efficient detection function, and the optimization is able to take advantage of that to produce higher detection probabilities with the same amount of effort. Qualitatively, the plans look similar.

Computing the T-optimal plan (in MATLAB) for the inverse cube detection function with decreasing-rate algorithm took about 100 times longer than with algorithm for the exponential detection function given in Chap. 3. For comparison, we also applied the MATLAB optimization function “fmincon” to the problem in this example. The function fmincon obtained an answer very similar to one found here, but it took 10 times as long as the algorithm for a decreasing rate detection function. The advantage of fmincon is that is a standard optimization routine and does not have to be handcrafted as the decreasing-rate algorithm does. For a one-time use, a standard optimization function such as fmincon might be preferred in spite of its longer run time.

5.3.4 Optimal Multi-type Search

This section presents an extension of the optimal detection problem to multi-type search. In a multi-type search, the target’s state is (i, x) where $i \in I$ is the target type and x is the target position in the standard two or three-dimensional space. In a search and rescue problem at sea, possible types of targets could be

$$i = 1, \text{ live person in raft}; \quad i = 2, \text{ live person in water}; \quad i = 3, \text{ dead person}. \tag{5.35}$$

The sweep width for the target depends on type as well as position and time. In particular, if the person is in a raft, the sweep width for visual detection is substantially larger than if he is in the water. If the person is dead, we can set the sweep width equal to 0 so that maximizing detection probability is equivalent to maximizing probability of detecting the target alive as was noted in Sect. 3.2.5.

The extension to multi-type search is handled by adding a component corresponding to target type to the state space $\mathcal{X}$. Specifically we set $\mathcal{X} = I \times S$ where I is the set of target types and S is the set of possible target positions. The sweep width W can depend on i as well as x and t. However, the search allocations depend only on the x component of target state. That is, we cannot allocate effort to position and target type, only to position. A target path ω specifies both target type and position over time, and target type can change over time on a path. It is convenient to write $\omega_t = (i(\omega_t), x(\omega_t))$ so that target type and position at time t are explicitly indicated. The definition of ζ in (5.3) becomes

$$\zeta(f, \omega, t) = \sum\nolimits_{s=0}^{t} W\left(i(\omega_s), x(\omega_s), s\right) f(x(\omega_s), s) \text{ for } \omega \in \Omega, \ \ t = 0, \ldots, T.$$

Define

$$\widehat{\beta}_f(i, x, t, z) = E_{ixt}\left[b\left(W(i, x, t)\, z + \sum_{s \neq t} W(i(\omega_s), x(\omega_s), s) f(x(\omega_s), s)\right)\right] \tag{5.36}$$

and

$$\begin{aligned}\widehat{\beta}'_f(i, x, t, z) &\\ &= E_{ixt}\left[b'\left(W(i, x, t)\, z + \sum_{s \neq t} W(i(\omega_s), x(\omega_s), s) f(x(\omega_s), s)\right)\right] W(i, x, t) \\ &\qquad\qquad \text{for } i \in I, x \in S \text{ and } z \geq 0\end{aligned} \tag{5.37}$$

where $E_{ixt}[\,]$ indicates expectation conditioned on the target being at position x and of type i at time t. For convenience we write $x \in S$ in place of the more precise but cumbersome notation x such that $(i, x) \in \mathcal{X}$.

Let $p_t(i, x) = \Pr\{X(t) = (i, x)\}$. Then (5.22) and (5.23) become

$$\beta_f(x, t, z) = \sum_{i \in I} \widehat{\beta}_f(i, x, t, z)\, p_t(i, x) \tag{5.38}$$

and

$$\beta'_f(x, t, z) = \sum_{i \in I} \widehat{\beta}'_f(i, x, t, z)\, p_t(i, x) \quad \text{for } x \in S, \ t = 0, \ldots, T, \ z \geq 0. \tag{5.39}$$

If b is a decreasing-rate detection function, then $\beta_f(x, t, z)$ as given in (5.38) will be one also, and the necessary and sufficient conditions of Theorem 5.2 apply.

In the implementation of the decreasing-rate detection function algorithm, we must modify the computation of $\zeta_s(\widehat{\omega})$ and r' in step 4 as follows

$$\zeta_s(\widehat{\omega}) = \zeta(\widehat{\omega}) - W(i(\omega_s), x(\omega_s), s) f_k(x(\omega_s), s)$$

and

$$r'(j, s, z) = \sum_{i \in I} W(i, j, s) \sum_{\left\{\widehat{\omega}:\widehat{\omega}_s = (i,j)\right\}} \Pr\{\widehat{\omega}\} b'(W(i, j, s) z + \zeta_s(\widehat{\omega})). \tag{5.40}$$

The computation of $\zeta(\widehat{\omega})$ in step 7 of the Implementation of the T-Optimal Search Plan Recursion must be similarly modified to be

$$\zeta(\widehat{\omega}) = \zeta_s(\widehat{\omega}) + W(i(\widehat{\omega}_s), x(\widehat{\omega}_s), s) f_{k+1}(x(\widehat{\omega}_s), s). \tag{5.41}$$

Example: Exponential Detection Function In this example, we consider a multi-type search in which there are three target types as given in (5.35) and the detection function b is exponential. Since b is exponential, r' in (5.40) becomes

$$r'(j, s, z) = \sum_{i \in I} W(i, j, s) e^{-W(i,j,s)z} \sum_{\left\{\widehat{\omega}:\widehat{\omega}_s = (i,j)\right\}} \Pr\{\widehat{\omega}\} e^{-\zeta_s\left(\widehat{\omega}\right)} \tag{5.42}$$

and for the computation of the Washburn bound in step 8, $D(f_{k+1}, j, t)$ becomes

$$D(f_{k+1}, j, t) = r'(j, t, f_{k+1}(j, t)) \quad \text{for } j = 1, \ldots, J,\ t = 0, \ldots, T. \tag{5.43}$$

A distress call reports that at time $t = 0$ a person was either in the water or a raft at location $(0, 0)$. This position report is characterized by a circular normal uncertainty with mean $(0, 0)$ and standard deviation $\sigma = 1.0$ nm in any direction. The search planner estimates equal probability that the person is in the water or a raft. The raft will drift generally northward due to winds and currents at about 2 kn. The drift of the person in the water (PIW) is not affected by the wind but will drift northward due to the current at about 1 kn. As a result, the uncertainty in the location of the raft spreads about twice as fast as the uncertainty in the location of the PIW. There is a possibility that the person will die from exposure (hypothermia) if he is not found quickly enough. This possibility is modeled by the following transition mechanism using the states in (5.35). At time $t = 0$ the search object (target) is in state $i = 1$ or $i = 2$ each with probability 0.5. There are no transitions for the first 5 h. After that there is an exponential distribution on the time at which the person will die, i.e., transition to state $i = 3$. For the person in the raft this distribution has a rate of 0.001 deaths per hours. For the PIW the rate is 0.15 deaths per hour reflecting the fact that a person in the water is more likely to die from hypothermia than the one in a raft. There are no transitions out of state $i = 3$. There are 2 h of search by helicopter available at times 6, 12, and 18 h. The helicopter searches at 90 kn

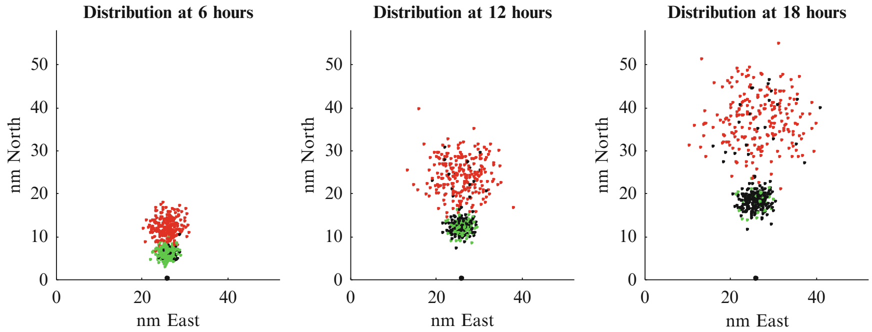


Fig. 5.4 Distributions of raft and PIW locations at times 6, 12, and 18 h. *Green* indicates PIW positions; *red* indicates raft positions. *Black* indicates particles that have died

yielding 180 nm of track length as the amount of search effort available at each of these times. The sweep width is 3 nm for the raft and 1 nm for the PIW reflecting the fact that it is easier to detect a raft than a PIW.

We wish to find a plan that maximizes the probability of detecting the person alive P_A rather than maximizing the probability of detection P_D, dead or alive. To do this we set the sweep width equal 0 for state 3 and calculate the T-optimal plan for this search. We call this the *optimal survivor search* plan.

Figure 5.4 shows the distributions for the raft and the PIW at times 6, 12, and 18 h represented by a sample of 500 particles from the 5000 equally weighted sample paths (particles) used for this example. The raft positions are indicated by red dots, the PIW by green dots. Black dots indicate particles that have died, i.e., transitioned to state 3. Figure 5.5 shows the fraction of particles alive in the two scenarios at times 6, 12, and 18 h.

Figures 5.6 and 5.7 show the optimal survivor search and the optimal detection search plans. We have $P_A = 0.44$ for the optimal survivor search compared to $P_A = 0.39$ for the optimal detection search. Observe that the optimal survivor search plan concentrates on the PIW in the early stages of the search when there is a higher probability of the PIW being alive. The optimal detection search expends most of it effort on the raft at 6 h before this distribution spreads out and then concentrates its search on the PIW. This produces a detection probability of 0.53 but a probability 0.39 of detecting the person alive which is more than ten percent below the probability for the optimal survivor search.

5.4 Optimal Plans for More General Payoff Functions

This section considers a class of problems with more general payoff functions that involve finding either discrete or continuous time allocation functions with search-like constraints that maximize the payoff function. These problems include

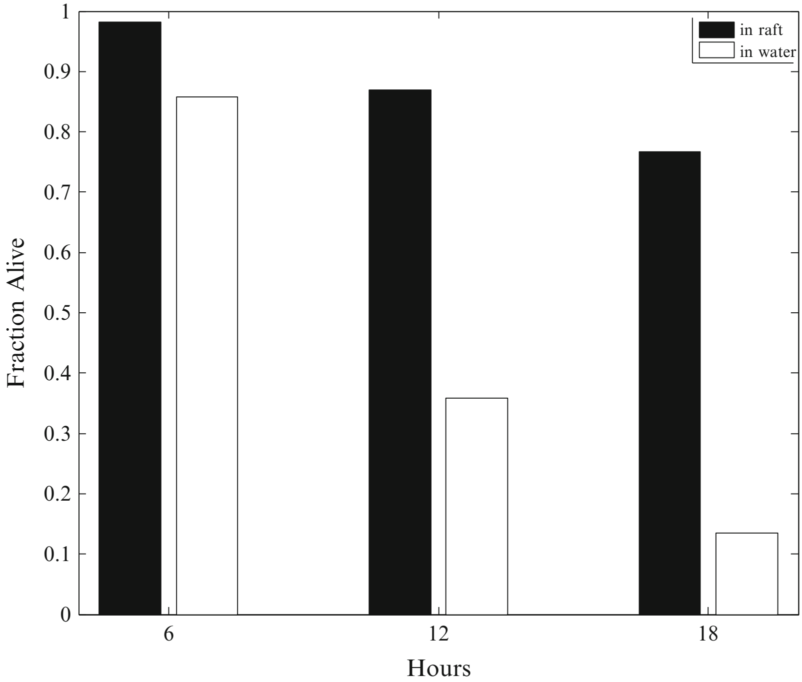


Fig. 5.5 Fraction of particles alive at 6, 12, and 18 h

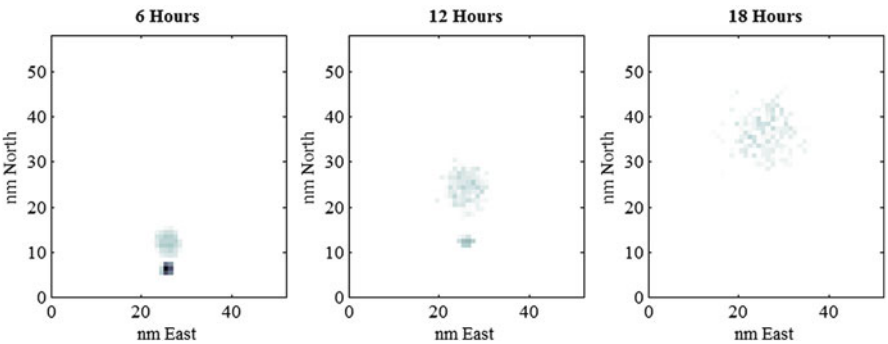


Fig. 5.6 Optimal survivor search plan yields $P_A = 0.43$. *Darker* colors indicate higher search densities

as special cases those considered in Sects. 5.1, 5.2 and 5.3. The concave FAB payoff functions defined in Chap. 3 are a special case of these more general payoff functions defined below. The results given in this section are a special case of those in Stromquist and Stone (1981).

The results and proofs in this section are more abstract and technical than those in the previous sections of this chapter and will involve Borel measurability

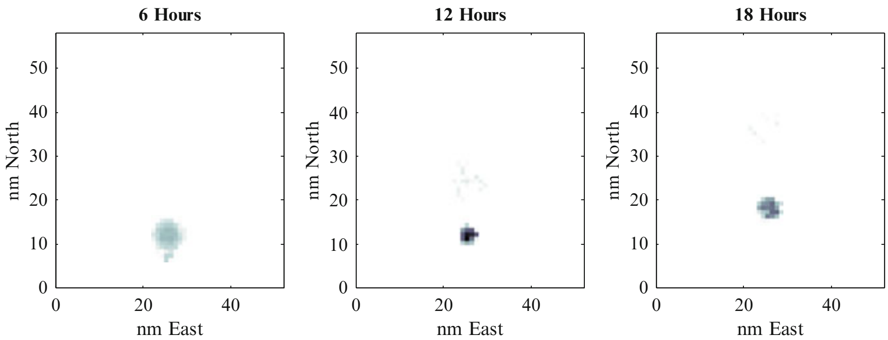


Fig. 5.7 Optimal detection search yields $P_A = 0.39$. *Darker* colors indicate higher search densities

assumptions. While Borel measurability is necessary for mathematical correctness, it imposes no constraint on functions that are used in applications. In particular all functions arising in applications will be Borel measureable.

5.4.1 Class of Problems

Let $[0, T]$ denote either the continuous interval of times from 0 to T or the set $t = 0, \ldots, T$. Let τ denote the measure on $[0, T]$. If $[0, T]$ is continuous, this is Lesbesque measure. If $[0, T]$ is discrete, τ is the counting measure on $\{0, \ldots, T\}$. If $a : [0, T] \to (-\infty, \infty)$ is a Borel measureable function, then

$$\int_0^T a(t) d\tau$$

will indicate either integration or summation depending on whether time is continuous or discrete.

Let F be the set of Borel measurable functions $f : S \times [0, T] \to [0, \infty)$. There is a Borel measureable cost function $c : S \times [0, T] \to (0, \infty)$ and constraint function $m : [0, T] \to [0, \infty)$. Define $F_B{}^+(m)$ to be the set of functions $f \in F$ such that

$$\begin{aligned} & f(x, t) \leq B \text{ for } x \in S,\ t \in [0, T] \\ & \int_S c(x.t) f(x, t)\, dx \leq m(t) \text{ for } t \in [0, T] \end{aligned} \tag{5.44}$$

and $F_B(m)$ to be set of $f \in F_B{}^+(m)$ such that

$$\int_S c(x.t) f(x, t)\, dx = m(t) \text{ for } t \in [0, T]. \tag{5.45}$$

We assume that B is large enough that (5.45) is satisfied for some $f \in {F_B}^+(m)$. Let $F^+(m) = F^+_\infty(m)$ and $F(m) = F_\infty(m)$.

Let U be a real-valued payoff functional defined on $F^+(m)$. An allocation $f^* \in F_B(m)$ is *optimal in* $F_B(m)$ if and only if

$$U\left(f^*\right) \geq U(f) \text{ for } f \in F_B(m).$$

The functional U is *concave* if and only if for any $f_1, f_2 \in {F_B}^+(m)$,

$$U\left((1-\varepsilon)f_1 + \varepsilon f_2\right) \geq (1-\varepsilon)\, U\left(f_1\right) + \varepsilon U\left(f_2\right) \quad \text{for} \quad 0 \leq \varepsilon \leq 1.$$

5.4.1.1 Gateaux Differential

For $f \in F^+(m)$ let $K(f)$ be the set of Borel measureable functions $h : S \times [0, T] \to (-\infty, \infty)$ such that $f + \varepsilon h \in F^+(m)$ for sufficiently small positive ε. Define the *Gateaux differential* of U at f in direction h by

$$U'\left(f, h\right) = \lim_{\varepsilon \to 0+} \frac{1}{\varepsilon}\left[U\left(f + \varepsilon h\right) - U(f)\right] \tag{5.46}$$

when the limit exists. Suppose there is a Borel measureable function $D\left(f, \cdot, \cdot\right) : S \times [0, T] \to (-\infty, \infty)$ such that for every $h \in K(f)$

$$U'\left(f, h\right) = \int_{t=0}^{T} \int_{S} D\left(f, x, t\right) h\left(x, t\right) dxdt. \tag{5.47}$$

Then $D\left(f, \cdot, \cdot\right)$ is the *kernel* of the Gateaux differential of U at f.

5.4.1.2 Upper Bound

For concave functionals that have a Gateaux differential with a kernel, there is a natural generalization of the Washburn bound. If U has a Gateaux differential U' at $f \in F_B^+$ with kernel $D\left(f, \cdot, \cdot\right)$, then for $t \in [0, T]$, we define

$$\overline{\lambda}\left(f, t\right) = \operatorname*{ess\ sup}_{x \in S} \frac{D\left(f, x, t\right)}{c\left(x, t\right)} \quad \text{and} \quad \underline{\lambda}\left(f, t\right) = \operatorname*{ess\ inf}_{\{x \in S\ : f_1(x,t) > 0\}} \frac{D\left(f, x, t\right)}{c\left(x, t\right)} \tag{5.48}$$

where we set $\underline{\lambda}\left(f, t\right) = 0$ if the measure of the set $\{x \in S : f\left(x, t\right) > 0\}$ is 0. Stromquist and Stone (1981) show that the functions $\overline{\lambda}\left(f, \cdot\right)$ and $\underline{\lambda}\left(f, \cdot\right)$ are Boreal measurable.

Theorem 5.3. *If U is a concave functional on $F_B(m)$ that has a Gateaux differential U' with kernel $D(f_1, \cdot, \cdot)$ at $f_1 \in F_B(m)$, then for any $f_2 \in F_B(m)$,*

$$U(f_2) - U(f_1) \leq U'(f_1, f_2 - f_1). \tag{5.49}$$

If in addition $\overline{\lambda}(f_1, t)$ and $\underline{\lambda}(f_1, t)$ are finite and non-negative for a.e. $t \in [0, T]$, then

$$U(f_2) - U(f_1) \leq \Delta(f_1) = \int_{t=0}^{T} \left(\overline{\lambda}(f_1, t) - \underline{\lambda}(f_1, t)\right) m(t)dt. \tag{5.50}$$

Proof Since U is concave,

$$\begin{aligned} U'(f_1, f_2 - f_1) &= \lim_{\varepsilon \to 0+} \tfrac{1}{\varepsilon}\left[U((1-\varepsilon)f_1 + \varepsilon f_2) - U(f_1)\right] \\ &\geq \lim_{\varepsilon \to 0+} \tfrac{1}{\varepsilon}\left[(1-\varepsilon)U(f_1) + \varepsilon U(f_2) - U(f_1)\right] \\ &= \lim_{\varepsilon \to 0+} \tfrac{1}{\varepsilon}\left[\varepsilon\left(U(f_2) - U(f_1)\right)\right] = U(f_2) - U(f_1) \end{aligned}$$

which proves the left-hand inequality in (5.49). Since U has a kernel at f_1, we have from (5.48) and left-hand inequality in (5.49) that

$$\begin{aligned} U(f_2) - U(f_1) &\leq \int_{t=0}^{T} \int_S D(f_1, x, t)\left(f_2(x, t) - f_1(x, t)\right) dxd\tau \\ &\leq \int_{t=0}^{T} \overline{\lambda}(f_1, t) \int_S c(x, t) f_2(x, t)\, dxd\tau \\ &\quad - \int_{t=0}^{T} \underline{\lambda}(f_1, t) \int_S c(x, t) f_1(x, t)\, dxd\tau \\ &\leq \int_{t=0}^{T} \left(\overline{\lambda}(f_1, t) - \underline{\lambda}(f_1, t)\right) m(t)d\tau = \Delta(f_1) \end{aligned}$$

which proves the theorem.

5.4.2 Necessary and Sufficient Conditions

In Theorems 5.4 and 5.5, we find necessary and sufficient conditions for $f^* \in F_B(m)$ to be optimal in $F_B(m)$. The proof of Theorem 5.5 employs measure theory concepts and considerations not addressed in previous chapters.

Theorem 5.4. *Suppose U is a concave functional defined on ${F_B}^+(m)$ having a Gateaux differential U' at $f^* \in F_B(m)$ with kernel $D(f^*,\cdot,\cdot)$. Then the following conditions are sufficient for f^* to be optimal in $F_B(m)$. There exists a Borel measurable function $\lambda : [0,T] \to [0,\infty)$ such that*

$$\begin{aligned} D(f^*,x,t) &\geq \lambda(t)c(x,t) \ \text{ for } f^*(x,t) = B \\ &= \lambda(t)c(x,t) \ \text{ for } 0 < f^*(x,t) < B \\ &\leq \lambda(t)c(x,t) \ \text{ for } f^*(x,t) = 0 \end{aligned} \tag{5.51}$$

for a.e. $(x,t) \in S \times [0,T]$.

Proof. By (5.49) and the assumption that U has a Gateaux differential U' at $f^* \in F_B(m)$ with kernel $D(f^*,\cdot,\cdot)$, we have for $f \in F_B(m)$

$$\begin{aligned} U(f) - U(f^*) &\leq U'(f^*, f - f^*) \\ &= \int_{t=0}^{T} \int_S D\left(f^*,x,t\right)\left(f(x,t) - f^*(x,t)\right) dx d\tau \\ &\leq \int_{t=0}^{T} \lambda(t) \int_S \left(c(x,t) f(x,t)\, dx - c(x,t) f^*(x,t)\right) dx d\tau = 0. \end{aligned} \tag{5.52}$$

The equality in the last line of (5.52) follow from $f^*,\ f \in F_B(m)$. The inequality follows from breaking the integral over S into the sum of integrals over S_1, S_2, and S_3 as follows.

$$S_1 = \{x : f^*(x,t) = B\},\ S_2 = \{x : 0 < f^*(x,t) < B\},\ S_3 = \{x : f^*(x,t) = 0\}.$$

On the set S_1, $D(f^*,x,t) \geq \lambda(t)c(x,t) > 0$ and $f(x,t) \leq f^*(x,t) = B$, so

$$\int_{S_1} D\left(f^*,x,t\right)\left(f(x,t) - f^*(x,t)\right) dx \leq \int_{S_1} \lambda(t)c(x,t)\left(f(x,t) - f^*(x,t)\right) dx.$$

On the set S_2, $D(f^*,x,t) = \lambda(t)c(x,t)$, so

$$\int_{S_2} D\left(f^*,x,t\right)\left(f(x,t) - f^*(x,t)\right) dx = \int_{S_2} \lambda(t)c(x,t)\left(f(x,t) - f^*(x,t)\right) dx,$$

and on the set S_3, $0 \leq D(f^*,x,t) \leq \lambda(t)c(x,t)$, and $f(x,t) \geq f^*(x,t) = 0$, so

$$\int_{S_3} D\left(f^*,x,t\right)\left(f(x,t) - f^*(x,t)\right) dx \leq \int_{S_3} \lambda(t)c(x,t)\left(f(x,t) - f^*(x,t)\right) dx.$$

This shows that (5.52) holds and proves sufficiency.

Theorem 5.5. *Suppose U is a functional defined on $F^{+}(m)$ having a Gateaux differential U' at $f^{*} \in F_B(m)$ with kernel $D(f^{*},x,t) \geq 0$ for a.e. $(x,t) \in S \times [0,T]$. Then the following conditions are necessary for f^{*} to be optimal in $F_B(m)$. There exists a Borel measurable function $\lambda : [0,T] \to [0,\infty)$ such that*

$$\begin{aligned} D(f^{*},x,t) &\geq \lambda(t)c(x,t) \ \text{ for } f^{*}(x,t) = B \\ &= \lambda(t)c(x,t) \ \text{ for } 0 < f^{*}(x,t) < B \\ &\leq \lambda(t)c(x,t) \ \text{ for } f^{*}(x,t) = 0 \end{aligned} \tag{5.53}$$

for a.e. $(x,t) \in S \times [0,T]$.

Proof. Let τ denote the measure on $[0,T]$ and μ the measure on S. Define

$$\lambda_l(f^{*},t) = \begin{cases} \operatorname*{ess\,inf}_{x\in S} \frac{D(f^{*},x,t)}{c(x,t)} & \text{if } \mu\{x : f^{*}(x,t) > 0\} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\lambda_u(f^{*},t) = \begin{cases} \operatorname*{ess\,sup}_{x\in S} \frac{D(f^{*},x,t)}{c(x,t)} & \text{if } \mu\{x : f^{*}(x,t) < B\} > 0 \\ \lambda_l(f^{*},t) & \text{otherwise.} \end{cases}$$

By Lemma 1 of Stromquist and Stone (1981), $\lambda_u(f^{*},\cdot)$ and $\lambda_l(f^{*},\cdot)$ are Borel measureable functions. Let $R = \{t : \lambda_u(f^{*},t) > \lambda_l(f^{*},t)\}$. We consider two cases, $\tau(R) = 0$ and $\tau(R) > 0$.

If $\tau(R) = 0$, we set $\lambda(t) = \lambda_u(f^{*},t) = \lambda_l(f^{*},t)$. If $f^{*}(x,t) = B$, the inequality in line 1 of (5.53) is satisfied because $\lambda(t) = \lambda_l(f^{*},t)$. If $0 < f^{*}(x,t) < B$, the equality is satisfied because $\lambda(t) = \lambda_u(f^{*},t) = \lambda_l(f^{*},t)$. If $f^{*}(x,t) = 0$, the inequality in line 3 of (5.53) is satisfied because $\lambda(t) = \lambda_u(f^{*},t)$. Thus conditions (5.53) are satisfied.

If $\tau(R) > 0$, we show that we can find an $f \in F_B(m)$ such that $U(f) > U(f^{*})$ which contradicts the assumption that f^{*} is optimal in $F_B(m)$. Since $D(f^{*},x,t) \geq 0$, we have $0 \leq \lambda_l(f^{*},t) < \infty$. Define

$$v(t) = \begin{cases} \frac{1}{2}(\lambda_u(f^{*},t) + \lambda_l(f^{*},t)) & \text{if } \lambda_u(f^{*},t) < \infty \\ \lambda_l(f^{*},t) + 1 & \text{if } \lambda_u(f^{*},t) = \infty \end{cases}$$

and

$$\begin{aligned} A &= \left\{(x,t) : t \in R,\ f^{*}(x,t) < B,\ \text{and } \frac{D(f^{*},x,t)}{c(x,t)} > v(t)\right\} \\ C &= \left\{(x,t) : t \in R,\ f^{*}(x,t) > 0,\ \text{and } \frac{D(f^{*},x,t)}{c(x,t)} < v(t)\right\}. \end{aligned} \tag{5.54}$$

Let $A_t = \{x : (x,t) \in A\}$ and $C_t = \{x : (x,t) \in C\}$. Both $\mu(A_t)$ and $\mu(C_t)$ are positive for $t \in R$. Let

$$\overline{c}(t) = \int_{C_t} c(x,t) f^*(x,t)\,dx,$$
$$\alpha(t) = \int_{A_t} c(x,t) \min\left\{B, \frac{\overline{c}(t)}{\mu(A_t)\,c(x,t)}\right\} dx \text{ for } t \in R.$$

Note that $\alpha(t) \le \overline{c}(t)$. Define

$$h(x,t) = \begin{cases} -f^*(x,t)\,\alpha(t)/\overline{c}(t) & \textit{for } (x,t) \in C \\ \min\left\{B, \frac{\overline{c}(t)}{\mu(A_t)c(x,t)}\right\} & \textit{for } (x,t) \in A \\ 0 & \text{otherwise.} \end{cases} \tag{5.55}$$

Note that

$$\int_S c(x,t)\,h(x,t)\,dx = 0 \text{ for } t \in R. \tag{5.56}$$

From (5.55) and (5.56) we have that for $0 \le \varepsilon \le 1$, $0 \le f^* + \varepsilon h \le B$ and $f^* + \varepsilon h \in F_B(m)$. Thus we may compute

$$\begin{aligned} U'(f^*,h) &= \int_0^T \int_S D\left(f^*,x,t\right) h(x,t)\,dx d\tau \\ &= \int_0^T \int_S \left(D\left(f^*,x,t\right) - v(t)c(x,t)\right) h(x,t)\,dx d\tau \end{aligned} \tag{5.57}$$

where the second equality follows from (5.56). The integrand in the second line of (5.57) is zero unless $(x,t) \in A$ or C. If $(x,t) \in A$, then

$$D\left(f^*,x,t\right) - v(t)c(x,t) > 0 \text{ and } h(x,t) > 0.$$

If $(x,t) \in C$, then

$$D\left(f^*,x,t\right) - v(t)c(x,t) < 0 \text{ and } h(x,t) < 0.$$

In either case the integrand is positive and $U'(f^*,h) > 0$. It follows from the definition of the Gateaux differential that for sufficiently small $\varepsilon > 0$, $U(f^* + \varepsilon h) > U(f^*)$ which contradicts the assumption that f^* is optimal in $F_b(m)$. Hence we must have $\mu(R) = 0$, and the theorem is proved.

5.4.3 Special Cases

One of the advantages of this more general approach, which uses Gateaux differentials, is that the sufficient conditions for exponential and decreasing rate detection functions are special cases of Theorem 5.4 and the necessary conditions, for $B = \infty$

are special cases of Theorem 5.5. The extension to multi-type target search is also a special case. In all these cases, the payoff function, probability of detection by time T, is a concave functional defined on $F_B(m)$ where $c(x,t) = 1$ for $(x,t) \in S \times [0,T]$. To find necessary and sufficient conditions for an optimal plan, one need only calculate the kernel function $D(f,\cdot,\cdot)$ for $f \in F_B(m)$ and apply the conditions of Theorem 5.4 or 5.5.

Decreasing Rate Detection Functions As an example, let us compute the Gateaux differential for $P(f,T)$ given by (5.5) when b is a decreasing rate detection function. In this case b has a bounded derivative, and we assume the sweep width $W(x,t)$ is bounded. Using the definition of Gateaux differential P' of P and the dominated convergence theorem to interchange expectation and limit, yields

$$\begin{aligned}
&P'(f,h)\\
&\quad= \lim_{\varepsilon\to 0+} \tfrac{1}{\varepsilon}\left\{P(f+\varepsilon h,T) - P(f,T)\right\}\\
&\quad= \lim_{\varepsilon\to 0+} \tfrac{1}{\varepsilon}\left\{E\left[b\left(\sum_{t=0}^{T} W(\omega_t,t)\left(f(\omega_t,t)+\varepsilon h(\omega_t,t)\right)\right)\right.\right.\\
&\qquad\qquad\left.\left.-b\left(\sum_{t=0}^{T} W(\omega_t,t) f(\omega_t,t)\right)\right]\right\}\\
&\quad= E\left[b'\left(\sum_{s=0}^{T} W(\omega_s,s) f(\omega_s,s)\right)\sum_{t=0}^{T} W(\omega_t,t)\, h(\omega_t,t)\right]\\
&\quad= E\left[\sum_{t=0}^{T} W(\omega_t,t)\, b'\left(\sum_{s=0}^{T} W(\omega_s,s) f(\omega_s,s)\right) h(\omega_t,t)\right].
\end{aligned}$$

Conditioning on $X(T) = x$, we obtain

$$\begin{aligned}
P'(f,h) &= \int E_{xt}\left[\sum_{t=0}^{T} W(\omega_t,t)\, b'\left(\sum_{s=0}^{T} W(\omega_s,s) f(\omega_s,s)\right) h(\omega_t,t)\right] p_t(x)dx\\
&= \sum_{t=0}^{T}\int_S E_{xt}\left[b'\left(\sum_{s=0}^{T} W(\omega_s,s) f(\omega_s,s)\right)\right] p_t(x) W(x,t)\, h(x,t)\, dx.
\end{aligned}$$

Thus

$$D(f,x,t) = E_{xt}\left[b'\left(\sum_{s=0}^{T} W(\omega_s,s) f(\omega_s,s)\right)\right] p_t(x) W(x,t) \quad \text{for } f \in F_B(m) \tag{5.58}$$

as in (5.8), and we see that the sufficient conditions (5.27) in Theorem 5.2 are a special case of Theorem 5.4 and the necessary conditions of Theorem 5.2 are a special case of the those in Theorem 5.5. However, the conditions of Theorems 5.4 and 5.5 apply to continuous as well as discrete time.

Notice that there are very few conditions on the stochastic process that models target motion. It does not have to be a diffusion process or a Markov process. What is required is that the sample paths be Borel measureable and the kernel of the Gateaux differential in (5.58) exist for a.e. $(x, t) \in S \times [0, T]$. Neither of these conditions is very restrictive. Note also that the necessary and sufficient conditions apply for any decreasing-rate detection function, not just an exponential one.

Exponential Detection Functions In the case where b is an exponential detection function, (5.58) becomes

$$\begin{aligned} D(f, x, t) &= E_{xt}\left[\exp\left(-\sum_{s=0}^{T} W(\omega_s, s) f(\omega_s, s)\right)\right] p_t(x) W(x, t) \\ &= e^{-W(x,t)f(x,t)} E_{xt}\left[\exp\left(-\sum_{s \neq t} W(\omega_s, s) f(\omega_s, s)\right)\right] p_t(x) W(x, t) \\ &= W(x, t)\, e^{-W(x,t)f(x,t)} q(x, t, f) \end{aligned} \tag{5.59}$$

which leads by Theorem 5.4 to the sufficient conditions

$$\begin{aligned} W(x, t)\, e^{-W(x,t)f^*(x,t)} q\left(x, t, f^*\right) &\geq \lambda_t c(x, t) \ \text{ for } f^*(x, t) = B \\ &= \lambda_t c(x, t) \ \text{ for } 0 < f^*(x, t) < B \\ &\leq \lambda_t c(x, t) \ \text{ for } f^*(x, t) = 0 \qquad \text{for } x \in S. \end{aligned} \tag{5.60}$$

The sufficient conditions of Theorem 5.1 follows readily from these conditions. Again the conditions in Theorem 5.1 apply to only discrete time whereas the conditions resulting from Theorems 5.4 and 5.5 apply to continuous and discrete time motion models.

Multi-Type Search If W depends on target type as in Sect. 5.3.4, then (5.58) becomes

$$D(f, x, t) = \sum_{i \in I} E_{ixt}\left[b'\left(\sum_{s=0}^{T} W(i(\omega_s), x(\omega_s), s) f(x(\omega_s), s)\right)\right] W(i, x, t)\, p_t(i, x) \tag{5.61}$$

from which we obtain $D(f, x, t) = \beta_f'(x, t, f(x, t))$ as defined in (5.39). As a result, the necessary and sufficient conditions for optimal multi-type search are extended to continuous as well as discrete time.

5.4.4 FAB Payoff Functions

In this section, we use Theorem 5.4 to extend the discrete time and space optimality conditions obtained in Theorem 3.5 for FAB payoff functions to continuous space.

As in Chap. 3,

$$\begin{aligned} &L(f,\omega) = a_{t+1}(\omega) + \sum\nolimits_{t=0}^{T} a_t(\omega)\, b\left(\zeta(\omega,t,f)\right) \text{ for } \omega \in \Omega \\ &\widehat{P}(f) = E\left[L(f,\omega)\right]. \end{aligned} \tag{5.62}$$

The difference from Chap. 3 is that the search space is continuous rather than discrete.

Corollary 5.6. *Suppose b is a decreasing-rate detection function. Then by setting* $c(x,t) = 1$ *and*

$$D\left(f^*,x,t\right) = W(x,t)\, E_{xt}\left[\sum_{s=t}^{T} a_s(\omega)\, b'\left(\zeta\left(\omega,s,f^*\right)\right)\right] p_t(x) \text{ for } x \in S,\ t=0,\ldots,T,$$

conditions (5.53) *become necessary for* $f^* \in F_B(m)$ *to satisfy* $\widehat{P}(f^*) \geq \widehat{P}(f)$ *for* $f \in F_B(m)$. *If in addition,* $L(f,\omega)$ *is a concave function of* f, *then conditions* (5.51) *are also sufficient for* $f^* \in F_B(m)$ *to satisfy* $\widehat{P}(f^*) \geq \widehat{P}(f)$ for $f \in F_B(m)$.

Proof. Computing the Gateaux differential of $\widehat{P}(f)$, we find that its kernel is

$$D(f,x,t) = W(x,t)\, E_{xt}\left[\sum_{s=t}^{T} a_s(\omega)\, b'\left(\zeta\left(\omega,s,f\right)\right)\right] p_t(x) \text{ for } x \in S,\ t=0,\ldots,T.$$

By Theorem 5.5, conditions (5.53) are necessary for $f^* \in F_b(m)$ *to satisfy* $\widehat{P}(f^*) \geq \widehat{P}(f)$ for $f^* \in F_B(m)$.

If $L(f,\omega)$ is a concave function of f, so is $\widehat{P}(f)$. Then by Theorem 5.4, conditions (5.51) are sufficient for $f^* \in F_B(m)$ to satisfy $\widehat{P}(f^*) \geq \widehat{P}(f)$ for $f \in F_B(m)$. This proves the corollary.

5.4.5 Optimal Resource Extraction with Uncertain Reserves

Theorems 5.4 and 5.5 can be applied to situations that are not related to search. Lipshutz and Stone (1992) applied versions of these theorems to find optimal resource extraction rates when the total amount of the resource available for extraction is uncertain. They considered the problem in both discrete and continuous time.

Discrete Time The amount A of resource available is uncertain. This uncertainty is modeled by a distribution function G where $\Pr\{A \leq a\} = G(a)$ for $a \geq 0$. For convenience, we will use

$$H(a) = 1 - G(a) \text{ for } a \geq 0.$$

Let $r(t, u, w)$ be the return we obtain when extracting the amount u of resource at time t given that w amount of resource has already been extracted. An extraction plan is given by a function $f: [1, T] \to [0, \infty)$ where $f(t)$ is the amount of resource extracted at time (period) t. There can be a bound B on the amount of resource extracted in single time period. Let δ be the discount rate on future income. The expected discounted return $R(f)$ for plan f is computed by

$$R(f) = \sum_{t=1}^{T} e^{-\delta t} r\left(t, f(t), Z(t-1)\right) H\left(Z(t)\right) \tag{5.63}$$

where

$$Z(t) = \sum_{s=1}^{t} f(s) \text{ for } t = 1, \ldots, T \text{ and } Z(0) = 0.$$

In (5.63) we assume that if the resource runs out during an extraction period, the return is 0 for that period. The objective is to find f^* to maximize R.

Let r_i and r_{ii} denote the first and second partial derivative of r with respect to the ith variable. We assume that $r(t, 0, w) = 0$, $r_2(t, 0, w) > 0$, and $r_{22}\cdot(t, u, w) < 0$ for $t = 1, \ldots, T$ and $u \geq 0$. The last two conditions imply that $r(t, \cdot, w)$ is concave and has decreasing rate of return. If R is concave, we can use Theorem 5.4 to obtain sufficient conditions for a plan f^* to be optimal. Since there is no cost constraint, we have $\lambda(t) = 0$ for $t = 1, \ldots, T$ in (5.51). Calculating the kernel of the Gateaux differential of R, we obtain

$$\begin{aligned}
&D(f, t, Z(t)) \\
&\quad = e^{-\delta t}\left[r_2\left(t, f(t), Z(t-1)\right) - r_3\left(t, f(t), Z(t-1)\right)\right] H\left(Z(t)\right) \\
&\qquad + \sum_{s=t}^{T} e^{-\delta s}\Big[r_3\left(s, f(s), Z(s-1)\right) H\left(Z(s)\right) + r\Big(s, f(t), Z(s-1) H'\left(Z(s)\right)\Big)\Big].
\end{aligned}$$

The following version of conditions (5.51) are sufficient for f^* to be an optimal extraction plan

$$\begin{aligned}
D(f^*, t, Z(t)) &\geq 0 \text{ for } f^*(t) = B \\
&= 0 \text{ for } 0 < f^*(t) < B \\
&\leq 0 \text{ for } f^*(t) = 0 \qquad \text{for } t = 1, \ldots, T.
\end{aligned} \tag{5.64}$$

Lipshutz and Stone present an algorithm for computing optimal extraction plans and give a number of examples of optimal plans under different assumptions on the bound B, discount rate δ, distributions on resources, and time varying rate of return function.

Continuous Time For the continuous time version of this problem, $r(t, u, w)$ becomes the instantaneous rate of return at time t when extracting at rate u and w is the resource that has already been extracted. The function f specifies a rate of extraction and

$$Z(t) = \int_{s=0}^{t} f(s)ds.$$

We seek to maximize

$$R(f) = \int_{0}^{T} e^{-\delta t} r\left(t, f(t), Z(t)\right) H\left(Z(t)\right) dt$$

In the special case where $B = T = \infty$, $r(t, u, w) = \rho(u)$ (i.e., r depends only on u), and $\rho'(0) = \infty$, Lipshutz and Stone use a version of Theorem 5.5 to prove the following result due to Loury (1978). The optimal extraction plan f^* satisfies

$$e^{-\delta t} \rho' \left(f^*(t)\right) = E\left[e^{-\delta t} \frac{\rho\left(f^*(\tau)\right)}{f^*(\tau)} \,\middle|\, \tau_R \geq t \right] \tag{5.65}$$

where τ_R is the random time at which the resource runs out using f^*. Equation (5.65) says that in an optimal plan, the discounted marginal rate of return at time t is equal to the expected value of the discounted average rate of return at the time extraction stops. Lipshutz and Stone also prove a more general version of this result.

5.5 Summary

We have extended the discrete time and space results of Chap. 3 to their analogous results for discrete time and continuous space in Sects. 5.1. 5.2 and 5.3. Section 5.3 presents an algorithm for calculating the T-optimal plan for a decreasing rate detection function. Section 5.4 considers a class of problems with more general payoff functions that involve finding either discrete or continuous time allocation functions with search-like constraints that maximize the payoff function. These problems include as special cases those considered in Sects. 5.1, 5.2 and 5.3 as well as extending them to continuous time. The FAB payoff functions defined in Chap. 3 are a special case of these payoff functions. Section 5.4 finds necessary and sufficient conditions for a function $f^* \in F_B(m)$ to be optimal.

5.6 Notes

Hellman (1972) obtained necessary conditions for an optimal search plan for a target whose motion is a diffusion process (Markov process with continuous sample paths) when the detection function is exponential. Saretsalo (1973) extended these necessary conditions to continuous time and space Markov processes, again when the detection function is exponential. Pershimo (1976, 1977) and Stone (1977) obtained necessary and sufficient conditions for a class of search problems called conditionally deterministic.

Brown (1977, 1980) applied the Karush-Kuhn-Tucker conditions to the problem of finding optimal plans for targets that move in discrete space and time when the detection function is exponential. By writing these conditions in a suitable form, he observed that the optimal plan for this moving target problem has the interesting property that if one selects a time t and conditions on failure at all times other than t (both before and after t), the optimal plan allocates the effort at time t so as to maximize the detection probability for the stationary target problem that one obtains from the conditioning. Since there are efficient methods for finding optimal plans for stationary targets, especially when the detection function is exponential, Brown was able to take advantage of this fact to devise an iterative algorithm that maximizes the probability of detecting the target in the interval $[0, T]$. This algorithm applies to target motions that are modeled by a mixture of discrete-time-and-space Markov chains and is very efficient (see Brown 1980). When search effort can be applied only in discrete looks and when detection on each look is independent of detection on any other look, Washburn (1980) gives an algorithm for finding search plans that satisfy a necessary condition for optimality. This algorithm is a discrete effort analog of Brown's algorithm.

In Stone et al. (1978), algorithms were devised for arbitrary discrete-time target motions and exponential detection functions. Stone (1979) generalized the necessary and sufficient conditions of Brown (1980) to target motions that are modeled by an arbitrary stochastic process with any mixture of discrete or continuous space or time. This generalization also applies to decreasing-rate detection functions.

All of the above results apply to the problem of finding a plan that maximizes the probability of detecting a target at time T, although Stone et al. (1978) also consider a payoff related to mean time to detection. In Stromquist and Stone (1981), the necessary and sufficient conditions for optimal detection search are generalized to a wider class of constrained optimization problems which include problems not related to search as well as numerous search-related ones. Washburn (1983) presents the FAB algorithm which is a generalization of the algorithms in Brown (1980) and Washburn (1980). FAB applies to finding optimal search allocations for payoffs in addition to maximizing the probability of detection by time.

The results and developments described above have two complementary sides, theoretical and practical. On the theoretical side is a set of necessary and often sufficient conditions for optimal plans for detecting moving targets and for more general optimization problems, specifically Brown (1980), Washburn (1980), Stone

(1979), and Stromquist and Stone (1981). On the practical side is a set of efficient algorithms for calculating optimal search plans for moving targets (i.e., Brown (1980); Stone et al. (1978); Discenza and Stone (1981); Washburn (1983)). Typically, these algorithms are developed from the necessary and sufficient conditions mentioned above. Benkoski et al. (1991) give an excellent review of the search literature as of 1991. Missing from this review are the developments since 1991 in optimal constrained path search, many of which are discussed in Chaps. 4 and 6.

References

Benkoski SJ, Monticino MG, Weisinger JR (1991) A survey of the search theory literature. Nav Res 38:469–494

Brown, SS (1977) Optimal and near optimal searches for a target with multiple scenario Markovian, constrained Markovian, or geometric memory motion in discrete space and time. Daniel H. Wagner, Associates Memorandum Report

Brown SS (1980) Optimal search for a moving target in discrete time and space. Oper Res 28:1275–1289

Discenza JH, Stone LD (1981) Optimal survivor search with multiple states. Oper Res 29:309–323

Hellman O (1972) On the optimal search for a randomly moving target. J SIAM Appl Math 22:545–552

Lipshutz RJ, Stone LD (1992) Optimal resource extraction with uncertain reserves. Nav Res 39:153–173

Loury GC (1978) The optimum exploitation of an unknown resource. Rev Econ Stud 45:621–636

Persiheimo U (1976) A control theory approach in the theory of search when the motion of the target is conditionally deterministic with stochastic parameters. Appl Math Optim 2:259–264

Persiheimo U (1977) On the optimal search for a target whose motion is conditionally deterministic with stochastic initial conditions on location parameters. SIAM J Appl Math 32:105–114

Saretsalo L (1973) On the optimal search for a target whose motion is a Markov process. J Appl Probab 10:847–856

Shapiro A, Dentcheva D, Ruszczynski L (2009) Lectures on stochastic programming. modeling, and theory. SIAM, Philadelphia

Stone LD (1979) Necessary and sufficient conditions for optimal search plans for moving targets. Math Oper Res 4:431–440

Stone LD (1977) Search for targets with generalized conditionally deterministic motion. SIAM J Appl Math 33:456–468

Stone LD, Brown SS, Buemi RP, Hopkins CR (1978) Numerical optimization of search for a moving target. Daniel H. Wagner, Associates Report to Office of Naval Research

Stromquist WR, Stone LD (1981) Constrained optimization of functionals with search theory applications. Math Oper Res 6:518–529

Washburn AR (1980) On search for a moving target. Nav Res 27:315–322

Washburn AR (1981) An upper bound useful in optimizing search for a moving target. Oper Res 29:1227–1230

Washburn AR (1983) Search for a moving target: the FAB algorithm. Oper Res 31:739–751

Chapter 6
Constrained Search in Continuous Time and Space

As we recall from Chap. 4, search applications might require the consideration of constraints on the agility of the searcher. For example, in the development of real-time controllers for autonomous systems it becomes essential to account for their often limited speed, turn radii, and other performance characteristics. When such systems need to be controlled on a very fine time scale, it becomes natural to formulate these problems in continuous time. Consequently, we are faced with the problem of optimal search in continuous time and space subject to constraints. This chapter provides an introduction to the subject through the formulation of several search situations as *uncertain optimal control problems.* Although there are computational challenges associated with the solution of uncertain optimal control problems, they are not unsurmountable. In fact, we include a section with examples that illustrate today's capabilities and demonstrate that practically useful solutions can be obtained in tens of minutes by standard optimization solvers. We also provide an introduction to the theory supporting such problems.

6.1 Formulation of Constrained Search Problems

We start with the formulation of constrained search problems in successively more complex situations. In Sect. 6.1.1, we consider a single searcher looking for a single target. We next consider multiple searchers and multiple targets in Sect. 6.1.2. Section 6.1.3 deals with finding an optimal trajectory for searchers patrolling a channel. We end in Sect. 6.1.4 with the statement of a generic optimal control problem that incorporates most of the previous cases as well as required approximations. The generic problem furnishes a framework for analysis pursued in a later section.

L.D. Stone et al., *Optimal Search for Moving Targets*, International Series in Operations Research & Management Science 237,
DOI 10.1007/978-3-319-26899-6_6

6.1.1 Search for a Single Target

We consider a planning horizon $[0, T]$ during which a target follows the stochastic process $X = \{X(t) : 0 \leq t \leq T\}$, with n-dimensional sample paths. Since X describes the "state" of the target, n might be larger than the dimension of the two- or three-dimensional physical space in which it operates. We assume that the underlying probability space is known such that a formulation in terms of paths is practically useful. That is, we let $\{X(t, \omega) : 0 \leq t \leq T\}$ be the path of the target under realization $\omega \in \Omega$ and assume that the probability measure P on the sample space Ω is given. In practice, Ω might contain N realizations specifying N possible target paths, perhaps generated by Monte Carlo simulation.

We seek to optimize the m-dimensional state $\{y(t) : 0 \leq t \leq T\}$ of a searcher that is looking for the target. Since we often consider velocity and other factors in addition to physical location, m is usually larger than two or three. The searcher is equipped with a sensor modelled by a nonnegative detection rate r that is defined such that $r(x, y)\Delta t$ approximates the probability of the searcher in state y detecting the target in state x during a small time interval $[t, t + \Delta t)$. The detection rate reflects the sensor effectiveness and we typically have that $r(x, y)$ is some decreasing function in the "distance" between x and y.

Next, we consider a particular realization $\omega \in \Omega$ and derive the probability of not detecting the target given its path $\{X(t, \omega) : 0 \leq t \leq T\}$ and a searcher trajectory $\{y(t) : 0 \leq t \leq T\}$. Let $q(t, \omega)$ be the probability that this trajectory fails to detect the target during $[0, t]$ given this target path. We assume that events of detection in non-overlapping time intervals are all independent, so we can calculate $q(t, \omega)$ recursively using the difference equation[1]

$$q(t + \Delta t, \omega) = q(t, \omega)\left(1 - (r(X(t, \omega), y(t))\Delta t + o(\Delta t))\right), q(0, \omega) = 1, \quad (6.1)$$

which becomes the parameterized differential equation

$$\dot{q}(t, \omega) = \frac{d}{dt}q(t, \omega) = -q(t, \omega)r(X(t, \omega), y(t)), \; q(0, \omega) = 1, \quad (6.2)$$

as $\Delta t \to 0$, with solution

$$q(t, \omega) = \exp\left(-\int_0^t r(X(s, \omega), y(s))ds\right). \quad (6.3)$$

[1]Recall that if a function f defined on the real line is $o(x)$, then $\lim_{x \to 0} \frac{f(x)}{x} = 0$.

Consequently, the probability that the searcher fails to detect the target during $[0, T]$, accounting for the uncertain ω, becomes

$$E[q(T,\omega)] = \int \exp\left(-\int_0^T r(X(t,\omega), y(t))dt\right) dP(\omega). \tag{6.4}$$

Our goal is to find a trajectory for the searcher that minimizes this expression. We assume that the motion of the searcher is governed by the differential equation

$$\dot{y}(t) = h(y(t), u(t)),\ t \in [0, T],\ y(0) = y_0, \tag{6.5}$$

where y_0 is the m-dimensional vector of initial conditions for the searcher and $u(t)$ is an m_u-dimensional vector of control input to the searcher at time t, which could be the rate of change of the heading of the searcher. The function h describes the response of the searcher to the control u. A search coordinator selects the initial condition y_0 within a given constraint set Y and control input $\{u(t) : 0 \leq t \leq T\}$ for the searcher, which for all times must be within a given constraint set U. Further constraints on the control input, for example depending on the current searcher state, are also possible, but omitted here; see for example Polak (1997, Chapter 4) and to a limited extent Sect. 6.1.3. Note that we are not directly optimizing the searcher's trajectory, as in previous chapters, but rather the inputs that determine it. The solution of (6.5), which we assume exists and is unique, then gives the state $\{y(t) : 0 \leq t \leq T\}$ of the searcher and an associated probability of failed search $E[q(T,\omega)]$ according to (6.4).

As for all infinite-dimensional problems, one needs to pay attention to the space in which the control is assumed to reside. At this point, we omit the technical details, which will be discussed in Sect. 6.3.

Given closed and convex constraint sets U and Y for controls and initial conditions, respectively, the *constrained search problem* takes the form

$$\begin{aligned} &\text{find control } u(t) \in U,\ t \in [0, T], \text{ and initial condition } y_0 \in Y \\ &\qquad \text{that minimize } E[q(T,\omega)] \\ &\qquad \text{subject to } \dot{y}(t) = h(y(t), u(t)),\ t \in [0, T],\ y(0) = y_0. \end{aligned}$$

The control constraints might reflect the maximum rate of change of heading that can be applied at a given point in time. The initial conditions could be fixed, reducing Y to a singleton, or reflect the possible locations from which the searcher might start its search. We notice that the objective function $E[q(T,\omega)]$ depends directly on the searcher trajectory $\{y(t) : 0 \leq t \leq T\}$ (see (6.4)), but only indirectly on the control input and initial condition through the differential equation (6.5). This search problem captures numerous applications and offers modeling flexibility through choices of detection rate r, target movement X, and searcher motion model h. Moreover, it provides a stepping stone towards generalizations involving multiple searchers and targets as seen next.

6.1.2 Search for Multiple Targets Using Multiple Searchers

The extension to multiple searchers and targets introduces additional notation, but otherwise involves few complications. We consider a collection of targets $\mathcal{K} = \{1, 2, \ldots, K\}$ as well as a group of searchers $\mathcal{L} = \{1, 2, \ldots, L\}$. The kth target follows the stochastic process $X^k = \{X^k(t) : 0 \leq t \leq T\}$, with n-dimensional sample paths. Again, we let P be the probability measure on the underlying sample space Ω, assumed common to all targets. We let $\{y^l(t) : 0 \leq t \leq T\}$ be the m-dimensional trajectory of the lth searcher. This searcher is equipped with a sensor characterized by the nonnegative detection rate $r^{k,l}$ against the kth target. As our notation emphasizes, searcher trajectories do not depend on the detection history; that is, we do not deal with situations where a searcher needs to follow (track) a target it finds for some time before it can proceed with search for other targets.

Following the development in Sect. 6.1.1, we find that given $\omega \in \Omega$, target path $\{X^k(t, \omega) : 0 \leq t \leq T\}$, and searcher trajectory $\{y^l(t) : 0 \leq t \leq T\}$, the probability that the lth searcher fails to detect the kth target during $[0, t]$ is

$$q^{k,l}(t, \omega) = \exp\left(-\int_0^t r^{k,l}(X^k(s, \omega), y^l(s))ds\right). \tag{6.6}$$

We assume that the searchers make independent detection attempts and can simultaneously detect multiple targets. Thus, the probability that no searcher detects any target during the time period $[0, T]$, given $\omega \in \Omega$, target paths $\{X^k(t, \omega) : 0 \leq t \leq T\}$ $k \in \mathcal{K}$, and searcher trajectories $\{y^l(t) : 0 \leq t \leq T\}$, $l \in \mathcal{L}$, is the product

$$\begin{aligned} &\prod_{k\in\mathcal{K}} \prod_{l\in\mathcal{L}} \exp\left(-\int_0^T r^{k,l}(X^k(t, \omega), y^l(t))dt\right) \\ &= \exp\left(-\int_0^T \sum_{k\in\mathcal{K}} \sum_{l\in\mathcal{L}} r^{k,l}(X^k(t, \omega), y^l(t))dt\right). \end{aligned}$$

The searchers might be linked in some manner and we therefore assume that they are governed by the potentially coupled dynamical system

$$\dot{y}(t) = h(y(t), u(t)),\ t \in [0, T],\ y(0) = y_0, \tag{6.7}$$

where

$$y(t) = (y^1(t), \ldots, y^L(t)) \text{ and } u(t) = (u^1(t), \ldots, u^L(t)) \tag{6.8}$$

are Lm-dimensional and Lm_u-dimensional vectors, respectively, with $u^l(t)$ being the control input at time t for the lth searcher. We assume that the differential equation has a unique solution for all (relevant) control input and initial conditions.

Again considering constraint sets U and Y, both assumed to closed and convex, the *constrained multi-searcher multi-target problem* that minimizes the probability that no searcher detects any target takes the form

$$\begin{aligned}
&\text{find control } u(t) \in U,\ t \in [0,T], \text{and initial condition } y_0 \in Y \\
&\qquad \text{that minimize } \int \exp\left(-\int_0^T \sum_{k\in\mathcal{K}} \sum_{l\in\mathcal{L}} r^{k,l}(X^k(t,\omega), y^l(t))dt\right) dP(\omega) \\
&\qquad \text{subject to } \dot{y}(t) = h(y(t), u(t)),\ t \in [0,T],\ y(0) = y_0.
\end{aligned}$$

An optimal solution of this problem tends to put emphasis on the targets that are most easily detected. The result is that "total failure," i.e., not a single detection, is made less likely. As an example of a situation where this might be desirable, consider the search for two insurgency leaders of the same organization. Although the capture of both leaders would have been ideal, the capture of one still allows for subsequent interrogations and possibly valuable intelligence gathering about the structure and facilities of the organization. Consequently, one might be willing to give up some chances for capturing both in return for increased probability of finding at least one.

Another possibility is to consider the expected number of targets detected as derived next. Given $\omega \in \Omega$, the probability that at least one searcher detects the kth target is simply

$$1 - \exp\left(-\int_0^T \sum_{l\in\mathcal{L}} r^{k,l}(X^k(t,\omega), y^l(t))dt\right).$$

Consequently, given $\omega \in \Omega$, the expected number of targets detected becomes

$$\sum_{k\in\mathcal{K}} \left\{1 - \exp\left(-\int_0^T \sum_{l\in\mathcal{L}} r^{k,l}(X^k(t,\omega), y^l(t))dt\right)\right\}.$$

The problem of maximizing the expected number of targets detected then takes the form

$$\begin{aligned}
&\text{find control } u(t) \in U, t \in [0,T], \text{and initial condition } y_0 \in Y \\
&\text{that maximize } \int \sum_{k\in\mathcal{K}} \left\{1 - \exp\left(-\int_0^T \sum_{l\in\mathcal{L}} r^{k,l}(X^k(t,\omega), y^l(t))dt\right)\right\} dP(\omega) \\
&\text{subject to } \dot{y}(t) = h(y(t), u(t)),\ t \in [0,T],\ y(0) = y_0.
\end{aligned}$$

In contrast to the previous formulation, an optimal solution here tends to spread the search effort across most of the targets. This objective function might be appropriate when searching for multiple drug smuggling boats as the total tonnage of drug captured is often the measure of success in counterdrug operations. Several other possible objective functions, for example involving groups of target types, are easily derived using similar principles.

6.1.3 Patrolling a Channel

A situation that allows for some specialization of the previous section, but also some extensions, is that of search for a single target that will pass straight down a channel at constant speed w some time in the future. The searchers are unaware of when the target will pass down the channel, which is assumed to be of constant width, but have some knowledge about the distance from the channel sides as specified by a probability density function. The goal is to construct trajectories for a group of searchers to be followed indefinitely, typically of the form back-and-forth across the channel, that minimize the probability that the target is not detected. The searchers are required to stay "near" a particular point in the channel and cannot venture up and down the channel. This problem traces its origin to World War 2 where the channel was the Straight of Gibraltar, searchers were Allied aircraft, and targets were German submarines. In this section, we formulate the channel patrol problem in a form similar to the constrained multi-searcher multi-target problem of Sect. 6.1.2 and make sure to account for the turn-radius and other performance characteristics of the searchers, which is critical in a narrow channel. As above, we let $y^l(t)$ be the m-dimensional state of the lth searcher at time t.

We derive the expression for the probability of no detection in two steps. First, we write the probability of no detection of a single stationary target during a fixed time period $[0, T]$. Second, we extend that expression to the situation at hand with a single moving target in a channel and an infinite time horizon. We end with remarks about a situation involving many targets.

We start by assuming, temporarily, that the target is stationary and located at $x = (x_1, x_2)$; an extension beyond two dimensions is routine but omitted here. Let $r^l(x, y^l(t))$ be the detection rate of the lth searcher at time t, given that the target is located at x. Then, similar to above, the probability that the lth searcher fails to detect the target during $[0, T]$ becomes

$$\int \exp\left(-\int_0^T r^l(x, y^l(t))ds\right)\phi(x)dx, \tag{6.9}$$

where ϕ is the (prior) probability density of the target location. The extension from a stationary target over a finite time horizon to a target that moves straight down a channel at constant speed and an infinite time horizon is accomplished by a linear transformation and other adjustments as described next.

We imagine the position of the target fixed on an infinitely long conveyor belt that moves down the channel at the speed of the target, w. Hence, the target is stationary relative to the conveyor belt and the formula derived above is applicable with minor changes in interpretation. Let $z^l(t)$ be the m-dimensional state of the lth searcher at time t relative to the conveyor belt. We adopt a coordinate system such that the channel is aligned in the direction of the second axis and the target moves down. We utilize subscripts to denote components of vectors. Thus, $y_1^l(t)$ is the first component of the state vector of the lth searcher at time t, which is assumed to represent the location of the searcher along the first axis, i.e., its placement across the channel. Moreover, $y_2^l(t)$ is the second component of the state vector and gives the location of the searcher along the second axis, i.e., its placement up and down the channel We assume that the searchers have limited freedom of movement up and down the channel so that $y_2^l(t)$ must be near zero for all t as specified below. Moreover, at time zero, the target has still not passed the searchers. Since the conveyor belt moves down at speed w, we have that for all $l \in \mathcal{L}$,

$$\begin{aligned} z_2^l(t) &= y_2^l(t) + wt \\ z_i^l(t) &= y_i^l(t), \text{for } i = 1, 3, 4, \ldots, m. \end{aligned} \tag{6.10}$$

For example, if the lth searcher is stationary at $y_1^l(t) = y_2^l(t) = 0$ in the channel for all t, it still moves along the conveyor belt in the vertical direction with $z_2^l(t) = wt$. We insist that each searcher carries out a trajectory that repeats itself every $T > 0$ time units, i.e., the state of the searcher at times $t = 0, T, 2T$, etc., are identical. This might be operationally convenient, but is also a framework that helps us overcome the challenges associated with an infinite time horizon. We refer to $y^l(t)$ and $z^l(t)$ as the absolute and relative states of the lth searcher at time t, respectively.

Suppose that the channel has width W and that we slice the conveyor belt into segments of length wT. The area of each segment is W times wT. Since the target is stationary relative to the conveyor belt, it will be located in a single segment for all $t \in [0, \infty)$. In contrast, the searchers will maintain $y_2^l(t)$ near zero, $l = 1, 2, \ldots, L$, and therefore will advance from segment to segment. Suppose that the target is located in the segment on which the searchers are also located during $[(k-1)T, kT]$ for a positive integer k, i.e., the target is located somewhere in the area $[0, W] \times [(k-1)wT, kwT]$. We assume that the searchers' sensors rapidly deteriorate as the distance between the target and the searchers increases. Thus, it suffices to only consider the time period $[(k-1)T, kT]$ and ignore the effect of search before and after this period when the searchers are anyhow well ahead of the target and well behind the target, respectively. In other words, despite the fact that the searchers will search indefinitely, they effectively have only "one shot" at detecting the target and that is during the time period when the target passes by their point of patrol. (In the case of a long-range sensor that violates this assumption, the expressed derived below is an upper bound on the probability of failed detection.) Hence, without loss of generality, we consider the time period $[0, T]$ and assume that the target passes the searchers during that time, i.e., the target is fixed to the "first" segment of the conveyor belt with coordinates in $[0, W] \times [0, wT]$. The location of the target on this segment is still unknown. We assume that there is a known probability density ϕ_1 on $[0, W]$ that quantifies the (prior) knowledge about the target's cross-channel location. However, there is no knowledge about when the target will pass the searchers. Thus, every along-channel location in $[0, wT]$ is as likely as any other.

In view of the above discussion, it follows that given a trajectory $\{z^l(t), 0 \le t \le T\}$, repeated indefinitely, the probability that the lth searcher fails to detect the target during time period $[0, \infty)$ is

$$\int_0^{wT} \int_0^W \exp\left(-\int_0^T r^l(x, z^l(t))dt\right) \phi_1(x_1)dx_1(1/wT)dx_2. \tag{6.11}$$

Since this formula is based on the consideration of only a single segment,

If the searchers have no prior knowledge of the across-channel position of the target, then one can assume a uniform distribution, i.e., $\phi_1(x_1) = 1/W$ for all $x_1 \in [0, W]$. Note that we abuse the notation r^l for detection rate slightly, by using it to represent the detection rate function both in the absolute and in the relative positions.

Since we assume that detection by one searcher is independent of detection by any other searcher, it follows straightforwardly that the probability that no searcher detects the target during time period $[0, \infty)$ is

$$\int_0^{wT} \int_0^W \exp\left(-\int_0^T \sum_{l \in \mathcal{L}} r^l(x, z^l(t))dt\right) \phi_1(x_1)dx_1(1/wT)dx_2. \tag{6.12}$$

As above, we assume that the searchers' motion satisfies the differential equation

$$\dot{y}(t) = h(y(t), u(t)),\ t \in [0, T],\ y(0) = y_0, \tag{6.13}$$

where again the vectors

$$y(t) = (y^1(t), \ldots, y^L(t)) \text{ and } u(t) = (u^1(t), \ldots, u^L(t)) \tag{6.14}$$

include (absolute) states and controls for all the searchers. Since the probability of no detection is given in terms of relative states $z(t)$, we translate these absolute dynamics into relative dynamics as follows. In view of the transformation (6.10), we find that (6.13) translates into

$$\begin{bmatrix} \dot{z}^1(t) - we \\ \dot{z}^2(t) - we \\ \vdots \\ \dot{z}^L(t) - we \end{bmatrix} = h\left(\begin{bmatrix} z^1(t) - wte \\ z^2(t) - wte \\ \vdots \\ z^L(t) - wte \end{bmatrix},\ u(t)\right)$$

where $e = (0, 1, 0, \ldots 0)^\top$ is the m-dimensional column vector that ensures that w and wt are subtracted from the second component of $\dot{z}^l(t)$ and $z^l(t)$, respectively. Thus, the dynamics of the searchers, in terms of the relative state, are concisely given as

$$\dot{z}(t) = g(z(t), u(t)), \; t \in [0, T], \; z(0) = z_0 = y_0, \tag{6.15}$$

where

$$g(z(t), u(t)) = h\left(\begin{bmatrix} z^1(t) - wte \\ z^2(t) - wte \\ \vdots \\ z^L(t) - wte \end{bmatrix}, \; u(t) \right) + \begin{bmatrix} we \\ we \\ \vdots \\ we \end{bmatrix}.$$

In contrast to the prior sections, where the search duration T is fixed, here it is natural to let the searchers optimize T as it is simply a parameter that determines the search trajectories. Recall that the search continues indefinitely and that T simply represents the cycle time of the trajectories. (The endurance of the searchers is not a factor as we assume, for example, that searchers are seamlessly replaced periodically.) A priori, it is unclear whether a large or small T is beneficial. Since the searchers are required to return to the same location every T time units, a small T might restrict the motion of the searchers such that, for example, they fail to search the edges of the channel. In addition, since we are only accounting for search taking place on the segment on which the target is fixed, a small T with a correspondingly short segment length might not be optimal as sensors' ability of looking up and down the channel might not be fully utilized. A large T might also be detrimental as the restrictions on motion up and down the channel could cause the searchers to follow intricate search patterns to avoid searching the same area excessively. The examples in Sect. 6.2.2 illustrate these effects.

The *optimal channel patrol problem* therefore takes the following form:

find control $u(t) \in U, t \in [0, T]$, initial condition $y_0 \in Y$, and time T that minimize

$$\int_0^{wT} \int_0^{W} \exp\left(- \int_0^T \sum_{l \in \mathcal{L}} r^l(x, z^l(t)) dt \right) \phi_1(x_1) dx_1 (1/wT) dx_2$$

$$\begin{aligned} \text{subject to } & \dot{z}(t) = g(z(t), u(t)), \; t \in [0, T], \; z(0) = z_0 = y_0 \\ & z(T) = \zeta(z_0) \\ & z^{\min}(t, T) \le z(t) \le z^{\max}(t, T) \; t \in [0, T] \\ & T^{\min} \le T \le T^{\max}, \end{aligned} \tag{6.16}$$

where $\zeta(z_0)$ imposes a requirement on the end state of z, possibly being simply $\zeta(z_0) = z_0$, $z^{\min}(t, T)$ and $z^{\max}(t, T)$ bound the state of the searchers, for example to remain within the channel and not to venture up and down the channel, and $T^{\min}$ and $T^{\max}$ limit the duration of a search cycle; see Sect. 6.2.2 for concrete instances.

In the optimal channel patrol problem, there is made no assumption about when the target will pass the searchers. If there is knowledge of that kind, for example that the target will pass between times t_1 and t_2, then it suffices to consider a conveyor belt of finite length and it becomes natural to let $T = t_2 - t_1$ or a multiple thereof. The situation then resembles that in Sect. 6.1.2, but with a single target.

In the case of multiple targets passing down the channel, again with no knowledge about when, it might be meaningful to minimize the fraction of targets not detected during $[0, \infty)$. If we assume that the targets are uniformly distributed over the conveyor belt according to a spatial Poisson process, then the optimal channel patrol problem again applies because the objective function can be reinterpreted as the fraction of targets not detected. In this case, that fraction relates to the ratio of the rate at which the searchers examine new area on the conveyor belt to the rate at which new area appears.

6.1.4 Optimal Control in an Uncertain Environment

It is beneficial to develop a generic problem that captures many practical situations and provides a vehicle for theoretical studies. In this section, we formulate such a problem, make connections with the concrete cases of Sects. 6.1.1, 6.1.2, and 6.1.3, and hint at a solution approach. Section 6.3 gives theoretical foundations for this problem.

We consider a probability space (Ω, Σ, P) and an uncertain dynamical system that for every $\omega \in \Omega$ is given by

$$\dot{x}(t, \omega) = h(x(t, \omega), u(t), \omega), \quad t \in [0, 1], \qquad x(0, \omega) = \xi + \iota(\omega), \tag{6.17}$$

where ξ is an n-dimensional vector representing the adjustable portion of the initial condition, $\{u(t) \ : \ 0 \leq t \leq 1\}$ is an m-dimensional control input, ι is the random portion of the initial condition, which can be thought of as noise, and h describes the behavior of the system. The resulting n-dimensional state is denoted by $\{x(t) \ : \ 0 \leq t \leq 1\}$. In this generic setting, x might represent the state of both targets and searchers, with usually a limited ability to "control" the targets through choice of u. We note that the formulation permits uncertainty in the searcher motion, which is often practically important as environmental conditions and dynamical models are often uncertain.

We aim to optimally select control u and the deterministic initial condition ξ such that the end state $x(1, \omega)$ is favorable on average as quantified by a real-valued function F. Specifically, the *optimal control problem under uncertainty* takes the form

$$\begin{aligned} &\text{find an initial state and control pair } \eta = (\xi, u) \text{ that minimize} \qquad (6.18)\\ &J(\eta) = \int F(x^{\eta}(1, \omega), \omega) dP(\omega)\\ &\text{subject to constraints } u(t) \in U \text{ for all } t \text{ and } \xi \in \Xi, \end{aligned}$$

where x^η is the solution of (6.17) under $\eta = (\xi, u)$, which is assumed to be unique, and U and Ξ are given compact and convex sets.

We note that the apparent limitation to a finite time horizon of $[0, 1]$ as well as an objective involving only the end time is not critical. It is well known that a problem on any time horizon and also a free-time problem can be converted to one on $[0, 1]$ by scaling time and other minor modifications. Moreover, a "running cost" is easily converted to a problem with only end-time cost by introducing an additional state; see for example Polak (1997, p. 493) and, in particular, Chung et al. (2011) for details about the optimal channel patrol problem. Consequently, the optimal control problem under uncertainty covers a wide range of search and related applications, including those involving uncertainty about the motion of the searchers. The probability P might therefore describe both uncertainty about targets as well as searchers.

The problem in (6.18) is typically only approachable through approximations since even the computation of the objective function for given η requires approximate numerical methods. We sketch an algorithmic framework that relies on Monte Carlo sampling for evaluation of the expectation in the objective function. This leads to a sequence of approximate *standard* optimal control problems that can be solved using well-develop algorithms.

We start by constructing a sequence of approximating problems. For a given sample size N, let $\{\omega^1, \omega^2, \cdots, \omega^N\}$ be an independent P-distributed sample. We approximate the objective function J by the sample average

$$J^N(\eta) = \frac{1}{N}\sum_{i=1}^{N} F\big(x^\eta(1, \omega^i), \omega^i\big). \tag{6.19}$$

Then, an approximate problem takes the form

$$\begin{aligned}&\text{find an initial state and control pair } \eta = (\xi, u) \text{ that minimize} \\ &J^N(\eta), \text{ subject to constraints } u(t) \in U \text{ for all } t \text{ and } \xi \in \Xi.\end{aligned} \tag{6.20}$$

The evaluation of $F\big(x^\eta(1, \omega^i), \omega^i\big)$ for all $i = 1, 2, \ldots, N$ is manageable since it involves the solution of the N dynamical systems

$$\dot{x}(t, \omega^i) = h(x(t, \omega^i), u(t), \omega), \quad t \in [0, 1], \qquad x(0, \omega^i) = \xi + \iota(\omega^i), \tag{6.21}$$

which are all subject to the same ξ and control input u. In fact, (6.20) is a standard optimal control problem for which there are many solvers such as DIDO (see Elissar Global 2015). These solvers often rely on discretization of time and the solution of large-scale nonlinear optimization problems; see for example Polak (1997, Chapter 4) and Ross and Karpenko (2012). Section 6.2 provides some details about such nonlinear optimization problems. In Sect. 6.3, we provide a glimpse into the theoretical foundations that justify solving (6.20) instead of the actual problem (6.18).

6.2 Examples of Constrained Search

We present two examples that illustrate solutions of constrained search problems. The first example considers a single searcher faced with a target following a complex motion model. The second example is a series of instances of the optimal channel patrol problem in Sect. 6.1.3. We follow the solution approach laid out above by approximating the expectation in the objective function of (6.18). In the first example, we use Monte Carlo simulation to construct this approximation and in the second example we use numerical integration. In both cases, this approach results in large-scale nonlinear optimization problems that are solved by SNOPT (Gill et al. 1998). Since these problems are nonconvex, the obtained search trajectories can only be expected to be locally optimal. The possibility of "poor" locally optimal solutions can be mitigated somewhat by selecting "good" initial trajectories and/or restarting the solver from multiple initial trajectories.

6.2.1 *Search for Target with Complex Motion*

A single searcher is attempting to detect a target moving across the rectangle $R = [-20, 20] \times [-10, 10]$ during the time horizon $[0, 75]$. The searcher has an imperfect sensor and a limited turn rate. Specifically, the searcher is assumed to travel with constant velocity $v = 1$ and the ability to change the heading with a rate of at most $u^{\max} = 0.25$ radians per unit time. Thus, the dynamics of the searcher are given by

$$\begin{aligned} \dot{y}_1(t) &= v \cos y_3(t), & \dot{y}_2(t) &= v \sin y_3(t), \\ \dot{y}_3(t) &= u(t) & |u(t)| &\leq u^{\max} \text{ for all } t \in [0, 75]. \end{aligned} \tag{6.22}$$

where (y_1, y_2) represents the position of the searcher, y_3 is the heading angle, and the control u is the turning rate.

We next turn to the movement of the target. The framework allows for nearly any target motion model and only requires that there is a mechanism for generating sample target paths. Here, we adopt a transparent but still relatively complex model with polynomial target paths. For each $\omega = (\omega_1, \omega_2, \cdots, \omega_{10})$, we define the 2-dimensional target path $X(t, \omega)$ by

$$X_1(t, \omega) = \omega_1 + \omega_2 t + \frac{1}{2}\omega_3 t^2 + \frac{1}{6}\omega_4 t^3 + \frac{1}{24}\omega_5 t^4,$$

$$X_2(t, \omega) = \omega_6 + \omega_7 t + \frac{1}{2}\omega_8 t^2 + \frac{1}{6}\omega_9 t^3 + \frac{1}{24}\omega_{10} t^4.$$

Let

$$A = \left\{ \omega \ : \ \omega_1 \in [0, 20], \omega_6 \in [-10, 10], \omega_2, \omega_7 \in \left[-\frac{1}{4}, \frac{1}{4} \right], \right.$$

$$\left. \omega_3, \omega_8 \in \left[-\frac{1}{40}, \frac{1}{40} \right], \omega_4, \omega_9 \in \left[-\frac{1}{800}, \frac{1}{800} \right], \omega_5, \omega_{10} \in \left[-\frac{1}{20{,}000}, \frac{1}{20{,}000} \right] \right\}$$

and

$$B = \{\omega \ : \ X(t, \omega) \in R \text{ for all } t \in [0, 75],$$
$$\dot{X}_1(t, \omega) < 0 \text{ for all } t \in [0, 75]\}.$$

Note that B is the set of ω for which the corresponding target path is in the rectangle R, and is moving from right to left, for all times $t \in [0, 75]$. The parameters in A are selected such that the target paths in a sample are reasonably different; see Fig. 6.1 for examples of paths. We set $\Omega = A \cap B$ and P the uniform distribution on Ω. This relatively complex target motion is not easily handled except by sampling, the approach we adopt below.

We model the detection rate using a Poisson Scan Model that reasonably well captures the performance of radar and sonar-based sensors; see Washburn (2014, Chapter 3) for details. Thus, the detection rate is given by

$$r(X(t, \omega), y(t)) = \beta \, \Phi\Big(\frac{F - D\|X(t, \omega) - (y_1(t), y_2(t))\|^2 - b}{\sigma} \Big), \tag{6.23}$$

where Φ is the standard normal cumulative distribution function, $\beta = 1$ is the scan opportunity rate, $F = 20$ is a sensor parameter, $\sigma = 10$ reflects the variability in the received signal strength, and $D\|X(t, \omega) - (y_1(t), y_2(t))\|^2 + b$ models the signal loss; see Figure 4.5 on page 93 of Wagner et al. (1999). As usual, $\| \cdot \|$ denotes the Euclidean norm. We use $b = 20$ and $D = 1$.

The problem then becomes that of minimizing

$$E\left[\exp\Big(-\int_0^{75} r(X(t, \omega), y(t)) dt \Big) \right], \tag{6.24}$$

subject to the dynamics (6.22), which is in the form of the constrained search problem in Sect. 6.1.1.

We construct an approximation with $N = 5000$ sample points drawn according to P. The resulting approximate problem is solved using an LGL-pseudospectral method with 54 nodes in the time domain; see for example Ross and Karpenko (2012) for algorithm details. This yields a $(5000 \cdot 54 = 270{,}000)$-dimensional

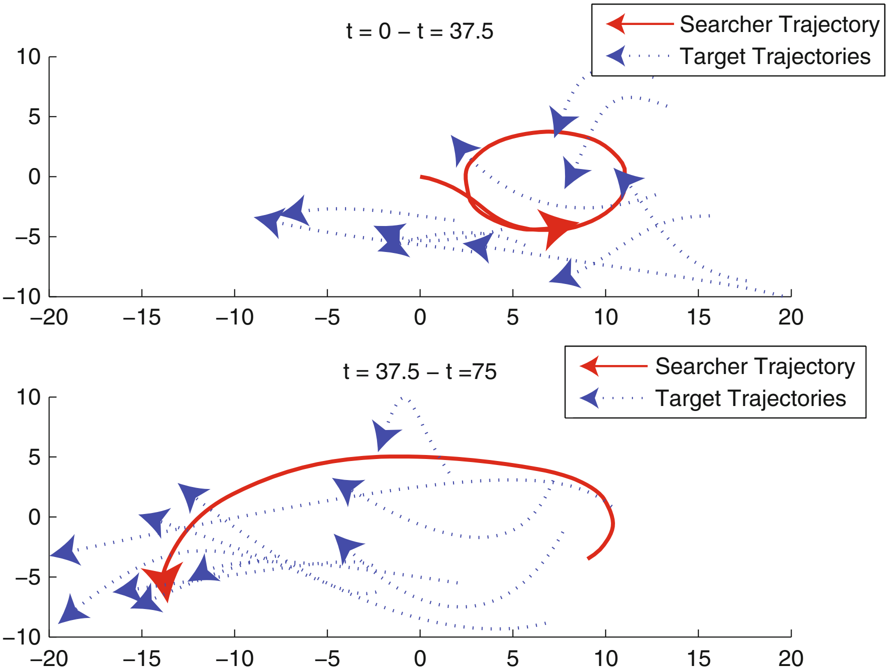


Fig. 6.1 Optimized trajectory (*solid line*) for a searcher attempting to detect a target following one of 5000 possible paths (10 illustrated by *dotted lines*). The searcher starts at $(0, 0)$ at time $t = 0$. The *arrows* in the figure indicate the direction of travel along the trajectories

nonlinear optimization problem. The total computing time is 35 min on an Intel Core i7-4700HQ laptop with 2.40 GHz and 16 GB RAM.

Figure 6.1 illustrates 10 of the 5000 target paths as dotted lines. The optimized searcher trajectory is given by the solid lines. The top portion of the figure gives the situation for the first 37.5 time units and the bottom portion for the remaining time. We see that the searcher, starting at the origin, which is not optimized, begins by moving to the right to "meet" the target, then turns around with the goal of "following" the target as it progresses from right to left.

6.2.2 Channel Patrol

We consider the optimal channel patrol problem defined in Sect. 6.1.3 and discuss several instances involving one, two, and three searchers.

6.2.2.1 Implementation Details

We consider searchers that each follow the dynamics of Sect. 6.2.1 with $u^{\max} = 1$ and $v = 1$. We utilize the formulation in (6.16) and ensure that the absolute location and heading of each searcher at time T is the same as at time 0. In the case of a single searcher, this is enforced by the function

$$\zeta(z_0) = (z_{0,1}, z_{0,2} - wT, z_{0,3} + 2n\pi)^\top \tag{6.25}$$

for some $n = 0, 1, 2, \ldots$, where $z_0 = (z_{0,1}, z_{0,2}, z_{0,3})^\top$ is the initial condition consisting of the location and heading of the searcher. Thus the constraint $z(T) = \zeta(z_0)$ guarantees that the searcher trajectory is closed. The integer n is a variable that determines the number of 360-degree rotations that are required during a patrol cycle and hence, as we will shortly see, partially determines the shape of the trajectory. For $n = 0$, the requirement becomes that any rotation must be associated with a corresponding rotation "back" as for example takes place when a searcher follows a figure-eight shape. The searcher might start with a clockwise rotation that is later compensated with equal counter-clockwise rotation. If $n = 1$, then the requirement becomes that the net total rotation must be 360 degrees as will be the case for a search that follows a racetrack-shaped trajectory. Ideally, we would have liked n to be freely determined by an optimization algorithm, but this would lead to mixed-integer programming and significant computational challenges. We overcome this issue by solving the problem for the various values $n = 0, 1, 2, \ldots$. In fact, it soon becomes apparent that one can expect the largest probability of detection for the values $n = 0, 1$; too many rotations become counterproductive.

We also impose constraints that ensure that the searchers remain within the width of the channel, i.e., horizontally in the interval $[0, W]$. Vertically, we ensure that the searchers do not venture too far up or down the channel and remain within the interval $[-\gamma, \gamma]$, for some $\gamma > 0$. We let the searchers start with any heading anywhere in the channel as long as the vertical coordinate is zero.

We set the channel width $W = 20$, where one unit of length equals 1000 yards, and the target speed $w = 3$. We assume that one unit of time equals 0.1 h. Hence, the target and searchers ($v = 1$) move at approximately 15 knots and 5 knots, respectively. Moreover, the limits on the control ensure that the searchers change their headings with at most one radian per 0.1 h. We always use $T^{\min} = 5$ and hence do not consider patrol cycles of shorter duration than 0.5 h. We vary $T^{\max}$.

We use the detection rate function (6.23) with parameters $\beta = 1$, $F = 70$, $b = 60$, $D = 0.5$, and $\sigma = 5$. If not stated otherwise, we assume that the distribution of the target's x_1-location is uniform, i.e., $\phi_1(x_1) = 1/W$.

Since the sample space Ω in this case is a rectangular subset of $\mathbb{R}^2$, we approximate the expectation in the objective function using numerical integration instead of by Monte Carlo sampling as the former is typically more accurate in low dimensions such as here. The resulting approximation again leads to a standard optimal control problem, which we solve using the Euler method for discretization and SNOPT for optimization; see details in Chung et al. (2011).

Next we describe the results of several numerical studies.

6.2.2.2 One Searcher

Table 6.1 provides numerical results for a single searcher, i.e., $L = 1$, for several values of the number of rotations n (see (6.25)), vertical trajectory constraint γ, and maximum patrol-cycle duration $T^{\max}$. In cases 1–3, $\gamma = W/10 = 2$, i.e., the searcher cannot move vertically more than two units above or below its starting point. Moreover, in cases 1–3, the patrol-cycle duration is limited to $T^{\max} = 25$. Case 1 requires the searcher to return to the same heading at the end of the patrol cycle (i.e., no rotation is allowed and $n = 0$ in (6.25)) forcing the optimized trajectory to have a "bow-tie" shape, as displayed in Fig. 6.2 (solid line). Figure 6.2 also displays the initial trajectory prior to optimization (dotted line). The arrows in Fig. 6.2 as well as all other figures indicate the direction of travel for the searcher. Large white and black triangles denote initial positions and headings before and

Table 6.1 Summary of numerical results for a single searcher and varying number of rotations n (see (6.25)), vertical range γ, and patrol-cycle duration limit $T^{\max}$. T^* and P^* are optimized patrol-cycle duration and probability of detection, respectively

Case	n	γ	$T^{\max}$	T^*	P^*
1	0	$W/10$	25	24.001	0.43348
2	1	$W/10$	25	23.568	0.43300
3	2	$W/10$	25	25.000	0.43243
4	0	$W/5$	15	15.000	0.42462
5	1	$W/5$	15	15.000	0.42620

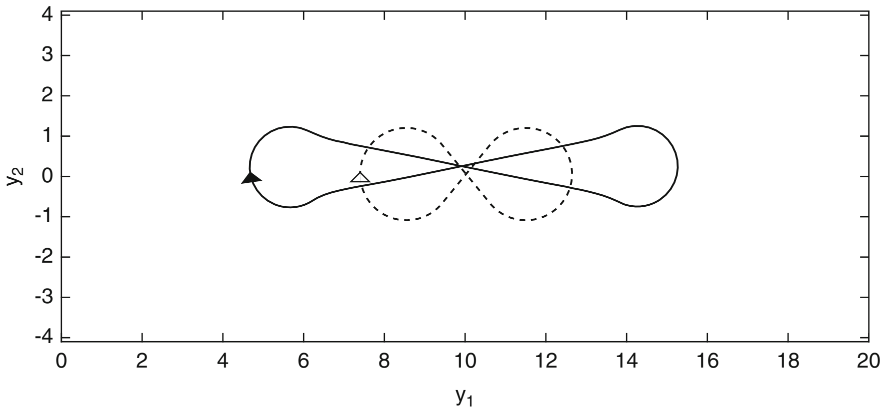


Fig. 6.2 Case 1: Initial trajectory (*dotted line*) and optimized trajectory (*solid line*) of a single searcher with no rotation ($n = 0$ in (6.25)). The *arrows* indicate direction of travel for the searcher. The *white triangle* denotes initial position and heading before the optimization, and the *black triangle* denotes the one after optimization

after optimization, respectively. Since the searcher's sensor range is roughly 5 units, the optimized trajectory is stretched out so that the sensor effectively reaches both sides of the channel. The initial trajectory has probability of detection 0.42145 and duration of a patrol cycle is 15, while the corresponding optimized numbers are 0.43348 and 24.001 as listed under T^* and P^* in Table 6.1. We note that in contrast to problems with a fixed time horizon, where usually it is optimal to search for the maximum allowed time, here we might very well prefer a patrol-cycle duration T that is strictly smaller than the maximum allowed $T^{\max}$. The reason is that we assume that the patrol cycle is repeated indefinitely and a patrol cycle of duration T is not assessed over time interval $[0, T^{\max}]$, but over $[0, T]$.

Figure 6.3 illustrates the "coverage" of the channel in Case 1. Specifically, it displays the probability of no detection at various relative locations along the conveyor belt. In the left portion of Fig. 6.3, giving the probabilities for the initial trajectory, large areas are not "covered" and thereby allowing the target a high chance of success. For the optimized trajectory (right portion), the situation is somewhat improved. In particular, the optimized trajectory makes it less likely that the target can slip through at the edges.

Case 2 in Table 6.1 is identical to Case 1 but requires a net total rotation of 360 degrees at the end of one patrol cycle (i.e., $n = 1$). Hence, the searcher must return

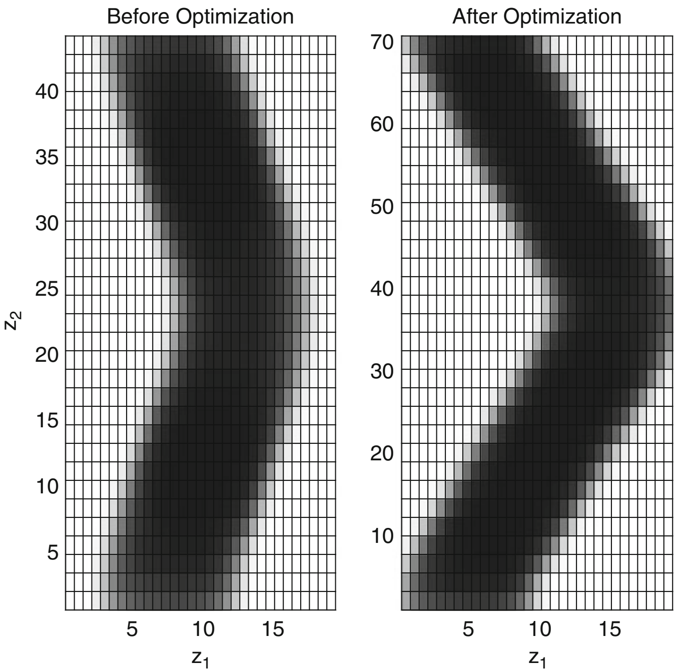


Fig. 6.3 Case 1: Coverage of channel before (*left*) and after (*right*) optimization in relative locations as measured by the probability of no detection. *Shades of gray* represent different probability levels with *black being* 0 and *white* 1

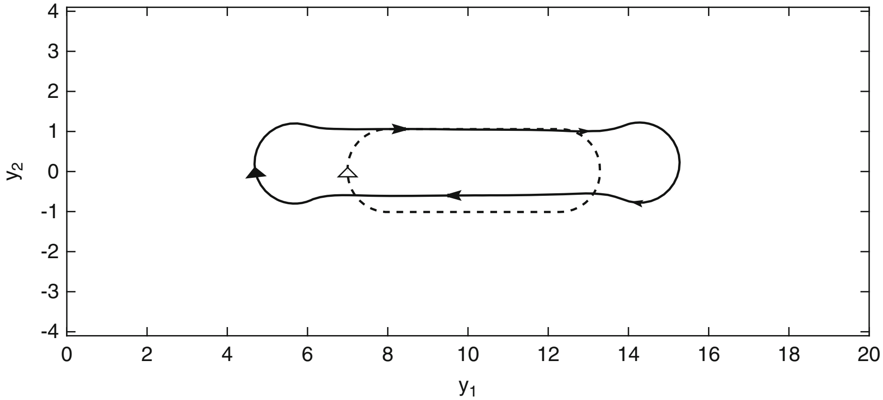


Fig. 6.4 Case 2: Initial trajectory (*dotted line*) and optimized trajectory (*solid line*) of a single searcher with 360-degree rotation ($n = 1$ in (6.25))

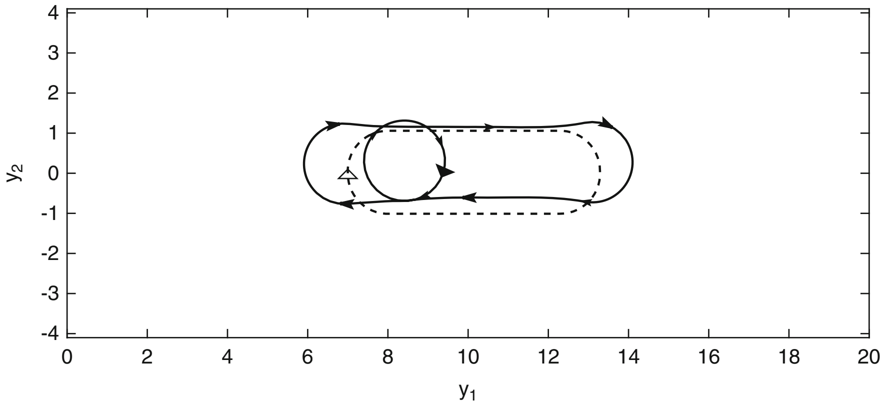


Fig. 6.5 Case 3: Initial trajectory (*dotted line*) and optimized trajectory (*solid line*) of a single searcher with 720-degree rotation ($n = 2$ in (6.25))

to a heading shifted 360 degrees from the initial heading, which excludes a "bow-tie" type trajectory, but is compatible with a "racetrack" type trajectory. Figure 6.4 shows the corresponding initial trajectory (dotted line, probability of detection is 0.42587) and optimized trajectory (solid line, probability of detection is 0.43300). We note that the optimized probability of detection is slightly worse for $n = 1$ than for $n = 0$, 0.43348 versus 0.43300.

Case 3 in Table 6.1 is identical to Case 1 but requires two rotations (i.e., $n = 2$), which rules out both "bow-tie" and "racetrack" type trajectories. In this case, the initial heading must be shifted by 720 degrees and hence the searcher makes two loops as shown in Fig. 6.5. (We note that the initial trajectory has $n = 1$.) The probability of detection is again slightly worse than for $n = 0$ and $n = 1$. Since the

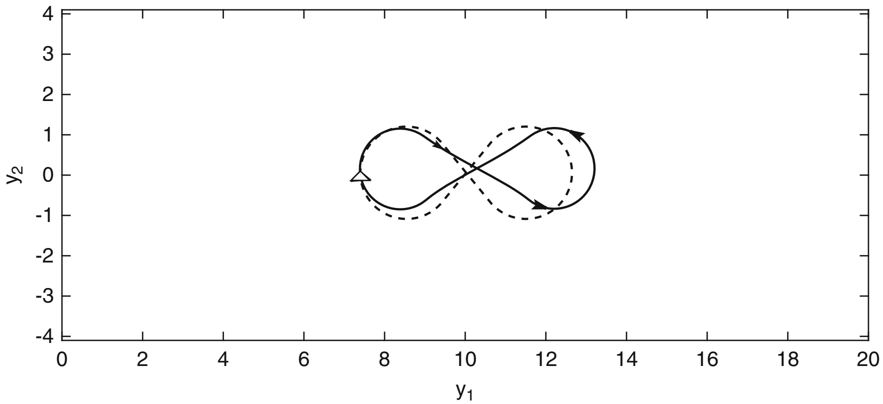


Fig. 6.6 Case 4: Initial trajectory (*dotted line*) and optimized trajectory (*solid line*) of a single searcher with no rotation ($n = 0$ in (6.25)) and restriction on patrol-cycle duration

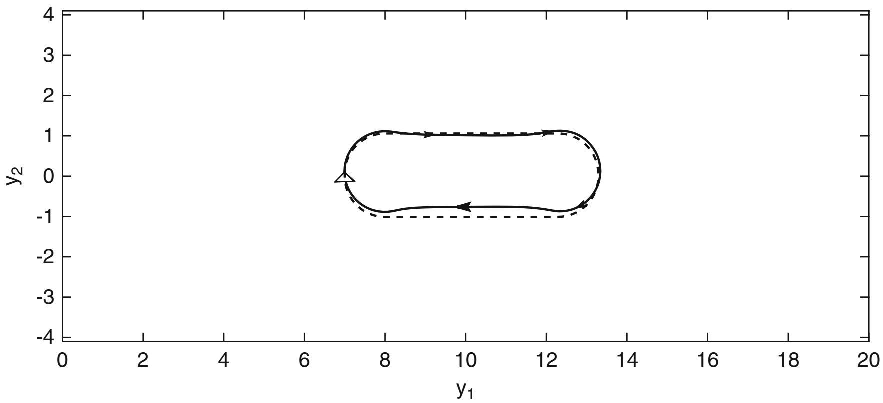


Fig. 6.7 Case 5: Initial trajectory (*dotted line*) and optimized trajectory (*solid line*) of a single searcher with 360-degree rotation ($n = 1$ in (6.25)) and restriction on patrol-cycle duration

probability of detection seems to decrease as the number of rotations increases, we will, as a heuristic, restrict ourselves to the problems with $n = 0$ and 1.

In Cases 1 and 2, the patrol-cycle duration limit $T^{\max}$ is not active. In Cases 4 and 5 this limit is reduced to 15 and also the vertical movement restriction γ is relaxed to $W/5 = 4$. We see from Table 6.1 that these changes impose a restriction on the searcher and the probability of detection worsens. Figures 6.6 and 6.7 show the resulting trajectories. We see that the worsened probability of detection is caused by the fact that the shorter patrol-cycle duration prevents the searcher from reaching the sides of the channel.

We also consider a situation (Case 6) where the distribution of the target's x_1-location is not uniform. Suppose that $\phi_1(x_1) = 2x_1/W$. Hence, we assume that the target is more likely to travel down the channel near the right side than the left side. Figure 6.8 shows the optimized trajectory for this case with no rotation

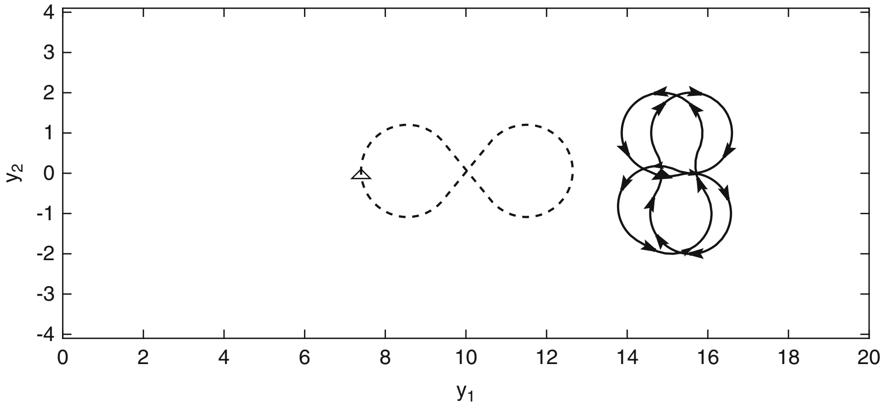


Fig. 6.8 Case 6: Initial trajectory (*dotted line*) and optimized trajectory (*solid line*) of a single searcher with no rotation ($n = 0$ in (6.25)) and right-leaning triangular target-location distribution

Table 6.2 Summary of numerical results for a single searcher, varying target speed w, and number of rotations n (see (6.25)), with $\gamma = W/10$ and $T^{\max} = 25$. T^* and P^* are optimized values of patrol-cycle duration and probability of detection, respectively

Case	w	n	T^*	P^*
1	3	0	24.001	0.43348
2		1	23.568	0.43300
7	2	0	23.578	0.49725
8		1	23.178	0.49514
9	1	0	24.177	0.65767
10		1	24.434	0.64077
11	0.5	0	25.000	0.88680
12		1	25.000	0.86413

required ($n = 0$), $\gamma = W/10$, and $T^{\max} = 25$. We see that in this case the searcher prefers a "double figure eight" trajectory close to the right side of the channel. The optimized trajectory has duration 25.000 and significantly improves the probability of detection to 0.61374 from the initial probability of detection of 0.42449.

We return to the situation with a uniform target distribution and consider the effect of variable target speed. Table 6.2 presents Cases 7–12 involving different target speeds and numbers of rotation. We assume that detection rate is as above, even though a slower target may be quieter and therefore harder to detect under certain circumstance. In all of these cases $\gamma = W/10$ and $T^{\max} = 25$. Rows two and three of Table 6.2 restate the results for Cases 1 and 2 from Table 6.1, in which the target speed $w = 3$, for ease of comparison. Rows four and five give results for $w = 2$. Naturally, as the target speed reduces, the probability of detection increases, while the shapes of trajectories remain qualitatively similar (Fig. 6.9). This effect is further observed for Cases 9 and 10 ($w = 1$) and for Cases 11 and 12 ($w = 0.5$). We note that in all cases the constraint of no rotation ($n = 0$) results in better probability of detection than the requirement of a 360-degree rotation ($n = 1$). These results are qualitatively different from the "idealized" results obtained in Wagner et al. (1999,

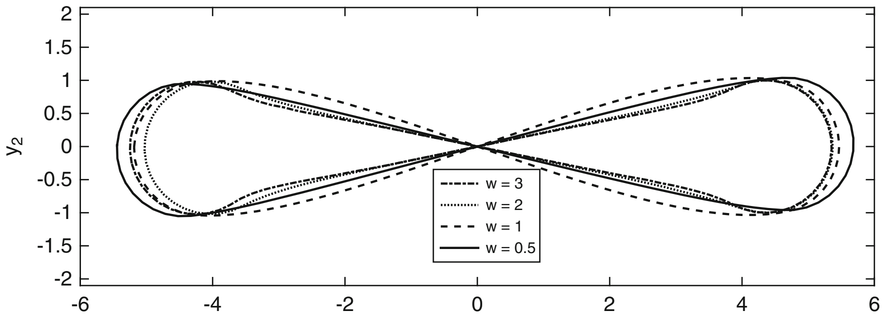


Fig. 6.9 Zoomed-in solution trajectories with varying w and $n = 0$ (see (6.25)). For ease of comparison, the trajectories are slightly translated so that the crossing points of the trajectories are at the origin

Table 6.3 Summary of numerical results for two searchers, varying number of rotations n (see (6.25)), and vertical range γ. T^* and P^* are the optimized patrol-cycle duration and probability of detection, respectively. For all cases in the table the patrol-cycle duration limit $T^{\max} = 25$

Case	n	γ	T^*	P^*
13	0	$W/10$	25.000	0.82037
14	1	$W/10$	11.633	0.79340
15	0,1	$W/10$	25.000	0.81234
16	0	$W/5$	25.000	0.82354
17	1	$W/5$	25.000	0.81594

Chapter 9), which do not account for turn radius constraints of the searcher. There we see that a "back-and-forth" trajectory similar to the one in Fig. 6.4 ($n = 1$), but with no constraints on the turn radius, is better than a "bow-tie" trajectory similar to that in Fig. 6.2 ($n = 0$) whenever v/w is less than 1.8. Since Cases 1, 2, 7–10 involve smaller v/w ratios, the "idealized" results would lead to the conclusion that a "back-and-forth" trajectory would be best. However, our numerical results show that the bow-tie trajectory ($n = 0$) is better when the searcher is constrained by its turn radius.

6.2.2.3 Two Searchers

Next we consider two searchers, i.e., $L = 2$, and five additional cases as summarized in Table 6.3. In all of these cases the patrol-cycle duration limit $T^{\max} = 25$. Rows two and three of Table 6.3 give the optimized patrol-cycle duration and probability of detection for no rotation ($n = 0$) and 360-degree rotation ($n = 1$), respectively,

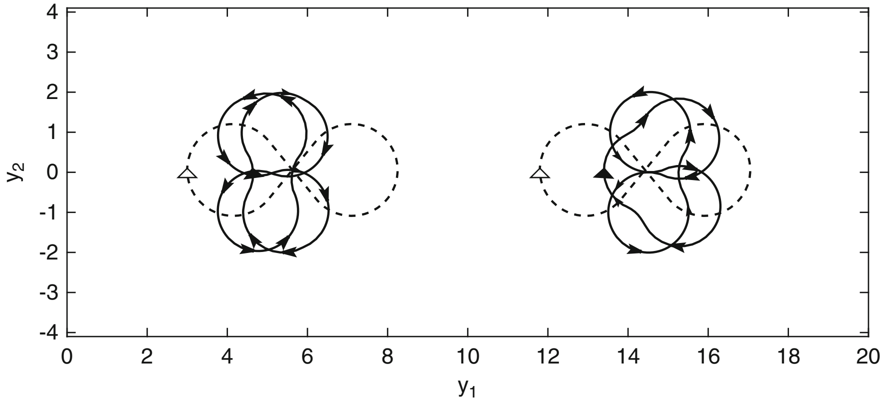


Fig. 6.10 Case 13: Initial trajectories (*dotted line*) and optimized trajectories (*solid line*) of two searchers with no rotation ($n = 0$ in (6.25))

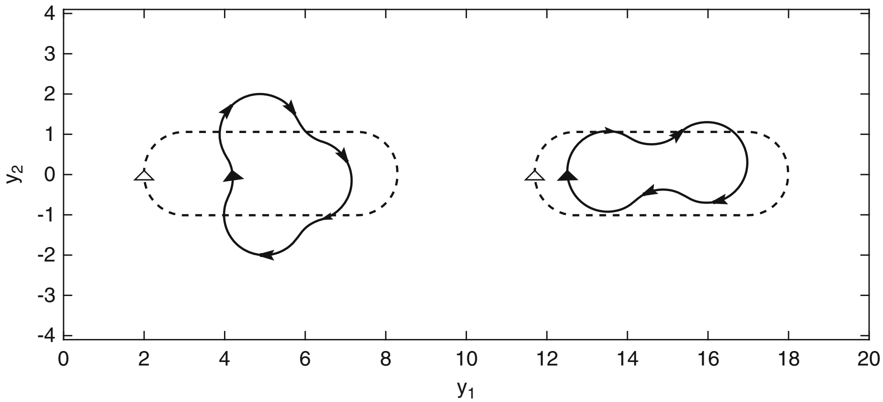


Fig. 6.11 Case 14: Initial trajectories (*dotted line*) and optimized trajectories (*solid line*) of two searchers with 360-degree rotation ($n = 1$ in (6.25))

using $\gamma = W/10$. Figures 6.10 and 6.11 give the corresponding trajectories. We see again that no rotation (Case 13) results in better probability of detection. Figure 6.10 shows that the optimized trajectories are similar to "figure eights," even though the initial trajectories are similar to the infinity symbol. This effect is caused by the narrowness of the channel. The two searchers obtain better probability of detection and less overlap in their "coverage" by moving along the channel instead of across. The probability of detection for the initial trajectory is 0.78003 and improves to 0.82037 after optimization.

We observe that the trajectories in Fig. 6.10 are different for the two searchers, which may be counterintuitive as the distribution of the target's x_1-location is uniform. Additional calculations show that the trajectories in Fig. 6.10 yield a larger probability of detection (0.82037) than patrol plans consisting of identical

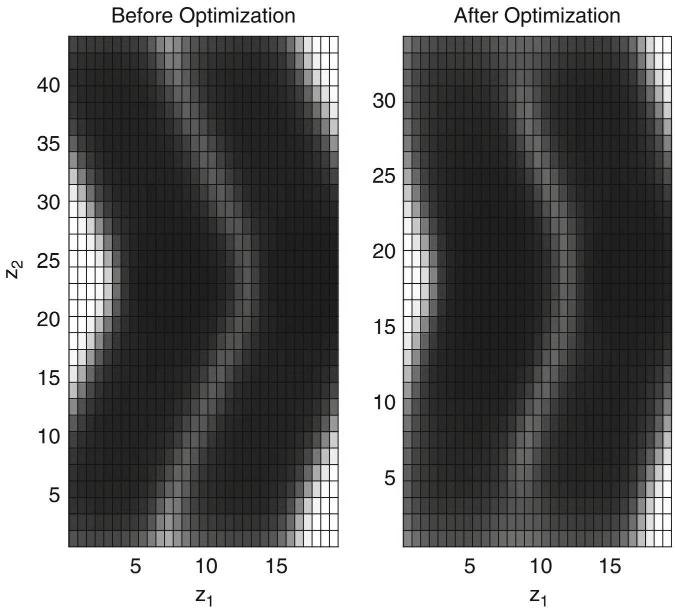


Fig. 6.12 Case 14: Coverage of channel before (*left*) and after (*right*) optimization similar to Fig. 6.3

but translated trajectories for both searchers. If the right-most searcher mimics the left-most searcher in Fig. 6.10, but on the right side of the channel, then the probability of detection deteriorates to 0.81630. If the left-most searcher mimics the right-most searcher, then the probability of detection deteriorates to 0.81472. The probabilities deteriorate further when the searchers carry out identical but mirror-imaged trajectories obtained by taking the trajectory of the left-most searcher and reconstructing its mirror image across the vertical line in the middle of the channel for the right-most searcher. These results provide new insight that is not easily obtained using the idealized calculations of Wagner et al. (1999, Chapter 9).

The optimized trajectories of Case 14 with the constraint of one rotation (i.e., $n = 1$) (see Fig. 6.11) yield a probability of detection of 0.79340, which is worse than in Case 13 (i.e., $n = 0$). Figure 6.12 illustrates the coverage of the channel in this case. We note that the initial trajectory (left portion of Fig. 6.12) leaves some locations poorly covered. The optimized trajectory is somewhat better in that regard as shown by the right portion of Fig. 6.12.

We also examined the configuration with one searcher constrained to no rotation ($n = 0$) and the other one to a 360-degree rotation ($n = 1$), and denote it by Case 15; see Fig. 6.13. However, the resulting probability of detection (0.81234) is worse than in Case 13.

Cases 16 and 17 in Table 6.3 show results similar to those for Cases 13 and 14, but for $\gamma = W/5$. With this relaxation ofthe vertical movement constraint for the searchers, we obtain slightly better probability of detection. The relaxation allows

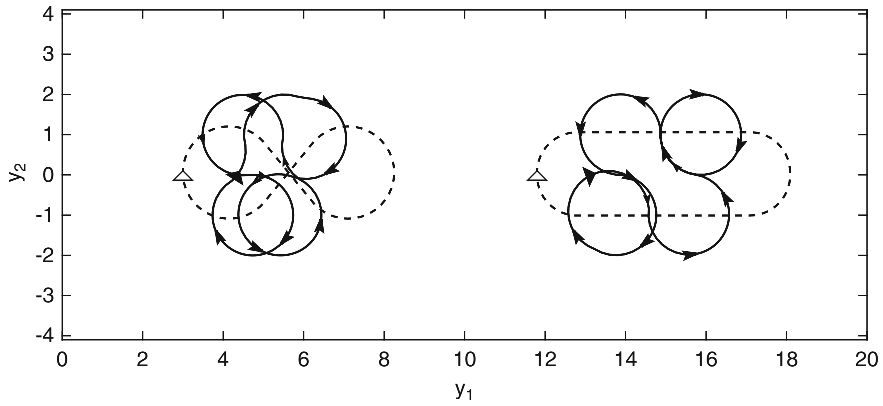


Fig. 6.13 Case 15: Initial trajectories (*dotted line*) and optimized trajectories (*solid line*) of two searchers with no rotation ($n = 0$) and one rotation ($n = 1$) in (6.25) and relaxed vertical trajectory constraint

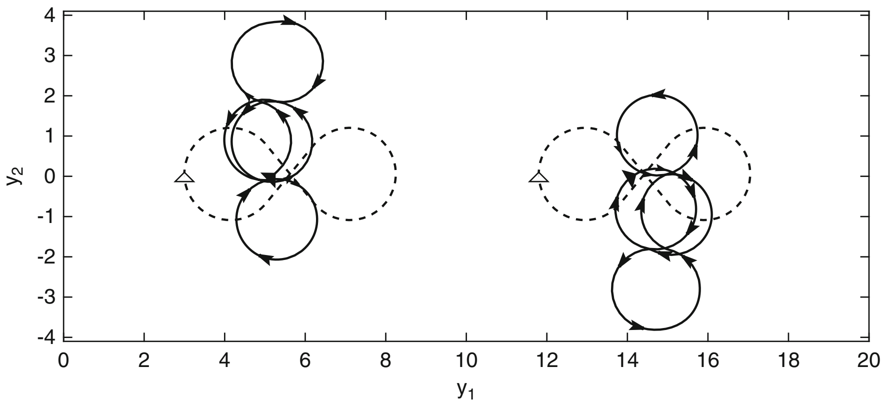


Fig. 6.14 Case 16: Initial trajectories (*dotted line*) and optimized trajectories (*solid line*) of two searchers with no rotation ($n = 0$ in (6.25)) and relaxed vertical trajectory constraint

for more complicated patrol trajectories as shown in Figs. 6.14 and 6.15. We see that the searchers stagger vertically their trajectories to avoid overlap and therefore increase the probability of detection. While not easily seen from Figs. 6.14 and 6.15, the searchers also synchronize their progress along their trajectories so that when one searcher moves to the left, say, then the other tends to move to the left also to fill the gap between the searchers. Figures 6.16 and 6.17 illustrate this effect by showing the coverage map and the relative locations of the searchers during $t \in [0, T]$ for Case 17, respectively. Such insight about the coordination between multiple searchers cannot be reached through single-searcher analysis. The initial trajectories in Case 17 result in a probability of detection of 0.77806, which is improved to 0.81594 after optimization.

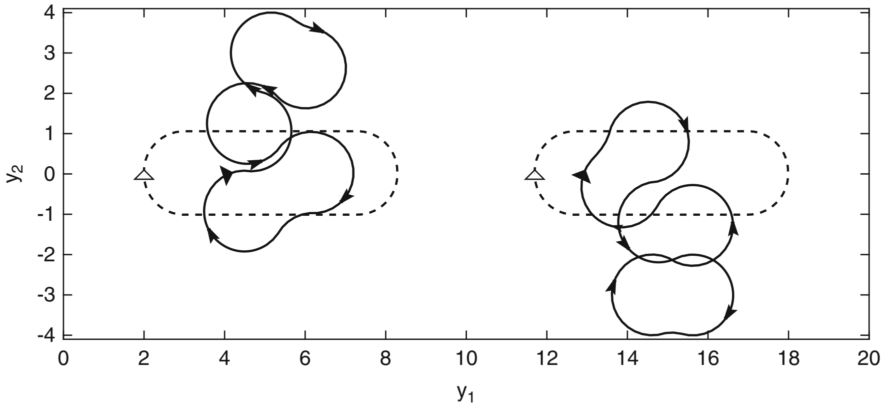


Fig. 6.15 Case 17: Initial trajectories (*dotted line*) and optimized trajectories (*solid line*) of two searchers with 360-degree rotation ($n = 1$ in (6.25)) and relaxed vertical trajectory constraint

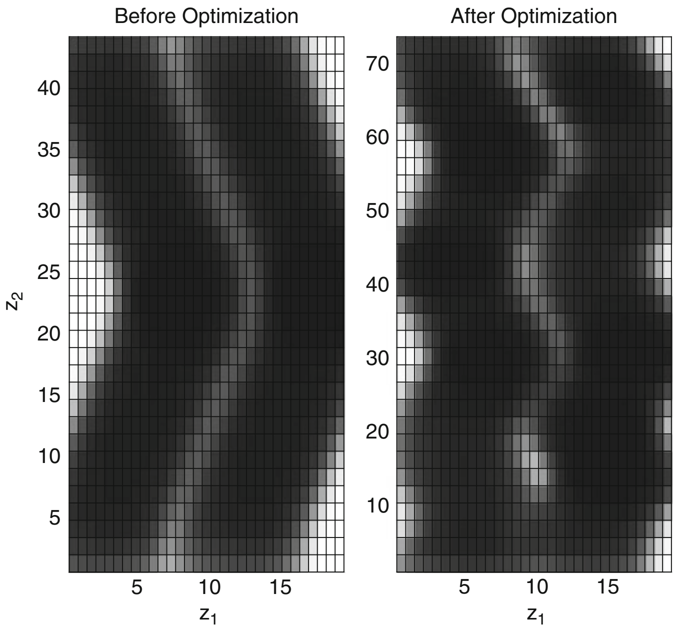


Fig. 6.16 Case 17: Coverage of channel before (*left*) and after (*right*) optimization similar to Fig. 6.3

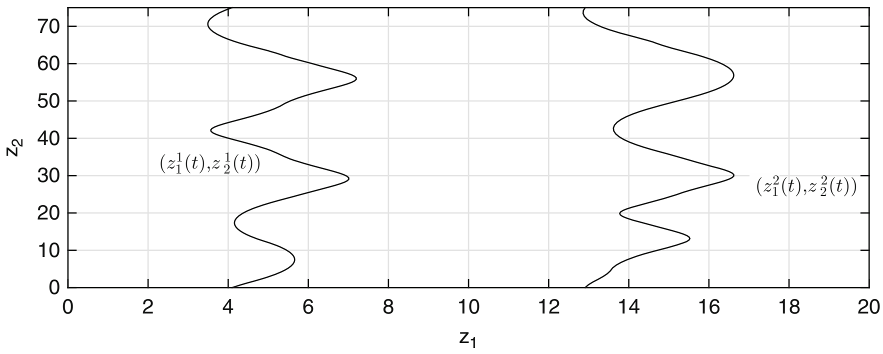


Fig. 6.17 Case 17: Relative locations for two searchers with absolute location given in Fig. 6.15

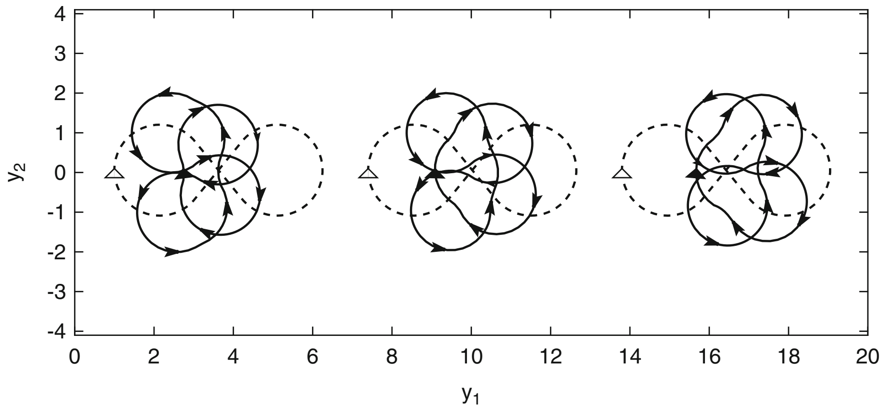


Fig. 6.18 Case 18: Initial trajectories (*dotted line*) and optimized trajectories (*solid line*) of three searchers with no rotation ($n = 0$ in (6.25)) constraint

6.2.2.4 Three Searchers

Finally, we consider three searchers briefly, for the single case of $T^{\max} = 25$, $\gamma = W/10$, and no rotation constraint ($n = 0$). The optimized probability of detection is 0.94086, improved from 0.90335 for the initial trajectories, and the optimized patrol-cycle duration is $T^* = 25.000$. Figures 6.18 and 6.19 display the initial and optimized trajectories in absolute locations and in terms of coverage, respectively. We see that the shape of each trajectory is quite similar to the ones in Case 13 for two searchers; see Fig. 6.10. We note that for two and three searchers the optimized trajectories tend to become quite intricate, especially when the searchers are tightly constrained vertically with $\gamma = W/10$ and no rotation is required ($n = 0$). This effect is caused by the fact that multiple searchers make it suboptimal for each searcher to search across the whole channel. This would have caused substantial overlap between the searchers and a lower probability of detection. Hence, each

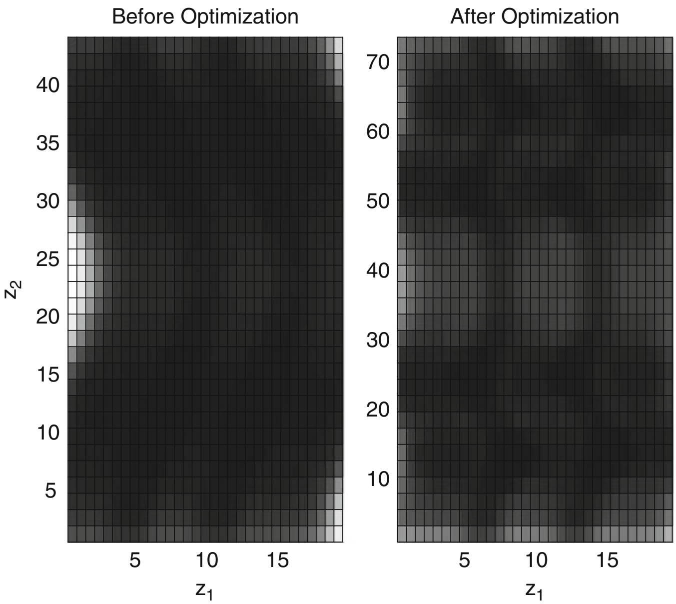


Fig. 6.19 Case 18: Coverage of channel before (*left*) and after (*right*) optimization similar to Fig. 6.3

searcher is effectively confined to a smaller area of operations. Even in the smaller area, the searchers tend to prefer longer patrol-cycle durations and the constraint $T \leq T^{\max}$ is often active. Longer patrol-cycle durations are usually preferable as the constraint that the searcher's relative final state must match its relative initial state (possibly with a rotational shift) imposes a restriction on the searcher and the longer duration allows more "free" movement between those "boundary conditions."

6.3 Theoretical Foundations

In this section, we summarize the theoretical foundations of a solution approach for (6.18) through the solution of the approximate problems (6.20). A rigorous treatment requires definition of spaces of controls and other technical details. Thus, we here assume that the reader is familiar with some functional analysis, but shy away from a comprehensive treatment such as those in Phelps (2015), Phelps et al. (2016), and Foraker et al. (2015a).

6.3.1 Preliminaries

We adopt the $\mathcal{L}_2$-topology and let $\mathcal{L}_2^m[0,1]$ be the space of all functions $v : [0,1] \rightarrow \mathbb{R}^m$ such that $\|v\|_2^2 = \int_0^1 \|v(t)\|^2 dt < \infty$, where $\|\cdot\|$ is the usual Euclidean norm. The initial-condition and control pairs are assumed to reside in a subspace of the Hilbert space

$$H_2 = \mathbb{R}^n \times \mathcal{L}_2^m[0,1],$$

where the inner product and norm on H_2 are defined for any $\eta = (\xi, u), \eta' = (\xi', u') \in H_2$ by

$$\langle \eta, \eta' \rangle_{H_2} = \langle \xi, \xi' \rangle + \langle u, u' \rangle_2,$$

with $\langle \xi, \xi' \rangle$ being the inner product for finite-dimensional vectors and $\langle u, u' \rangle_2 = \int_0^1 \langle u(t), u'(t) \rangle dt$. Therefore the norm in H_2 is given by

$$\|\eta\|_{H_2}^2 = \|\xi\|^2 + \|u\|_2^2.$$

The control $u(t)$ is constrained to be in a compact convex subset $U \subset \mathbb{R}^m$ for every $t \in [0,1]$ and the initial condition ξ is in a compact, convex sets $\Xi \subset \mathbb{R}^n$. The set of admissible controls is then

$$\mathbf{U} = \{u \in \mathcal{L}_2^m[0,1] \, : \, u(t) \in U \text{ for every } t \in [0,1]\}.$$

Thus, the set of all admissible state-control pairs for this problem is given by

$$\mathbf{H} = \Xi \times \mathbf{U}.$$

The set-up captures many practical applications including situations where the control input needs to be a discontinuous function of time as in the case of "bang-bang" control. More general constraints on u that depend on the state can be handled through penalties and incorporated into the objective function; see Polak (1997, Chapter 4) for an explicit treatment of such "state constraints." We refer to the optimal control problem under uncertainty given in (6.18) with these constraints as Problem OCPU.

6.3.2 Optimality Conditions

In this section we develop an optimality condition for Problem OCPU using an infinite-dimensional extension of the following classical result: if $x^* \in \mathbb{R}^n$ is optimal for the problem $\min f(x)$ subject to $x \in C$, with f continuously differentiable and C convex, then

$$\min_{x \in C} \nabla f(x^*)^\top (x - x^*) + (1/2)\|x - x^*\|^2 = 0; \tag{6.26}$$

see Polak (1997, Section 1.1.1). The first step in such a development is to find derivative expressions of the objective function with respect to the decision variable $\eta = (\xi, u)$, consisting of the initial condition and control pairs.

A set of mild assumptions needs to be impose to ensure that for each $\omega \in \Omega$ the dynamical system (6.17) in Problem OCPU has a unique solution $x^\eta(\cdot, \omega)$ with the terminal state $x^\eta(1, \omega)$ being continuously differentiable with respect to η. A bounded solution is guaranteed if there is no finite escape time for any ω, which holds for input-to-state stable systems and systems for which h in (6.17) is globally Lipschitz continuous or satisfies a linear growth condition in the state variable. Differentiability of h and the criterion function F in (6.18) with respect to control and states as well as integrability with respect to ω, especially of Lipschitz constants, ensure the required existence, uniqueness, and smoothness of the solution of the dynamical system (6.17) as well as smoothness in η of $F(x^\eta(1, \omega), \omega)$ for each ω; see Phelps et al. (2016) for details.

Focusing on the properties of the objective function in Problem OCPU as a function of the decision variable $\eta = (\xi, u)$, we let $\psi : \mathbf{H} \times \Omega \to \mathbb{R}$ be given by

$$\psi(\eta, \omega) = F(x^\eta(1, \omega), \omega).$$

Thus,

$$J(\eta) = E\big[\psi(\eta, \omega)\big] \text{ and } J^N(\eta) = \frac{1}{N}\sum_{i=1}^{N} \psi(\eta, \omega^i).$$

We proceed by finding expressions for the ($\mathcal{L}_2$-Frechet) derivatives of the objective functions J and J^N, and first state such derivatives for ψ with respect to its first argument. In the following, we use h_x and h_u for the Jacobian of h with respect to x and u, respectively. A subscript assigned to ∇ indicates the variables with respect to which a gradient is taken.

We then find that for any $\omega \in \Omega$, $\psi(\cdot, \omega)$ has a Gateaux differential given by

$$D\psi(\eta; \delta\eta; \omega) = \langle \nabla_\eta \psi(\eta, \omega), \delta\eta \rangle_{H_2}$$

at $\eta = (\xi, u)$, where $\delta\eta$ is the direction of change in decision. The gradient

$$\nabla_\eta \psi(\eta, \omega) = \big(\nabla_\xi \psi(\eta, \omega), \nabla_u \psi(\eta, \omega)\big)^\top$$

has two parts. The first part corresponds to derivatives with respect to initial conditions and is given by

$$\nabla_\xi \psi(\eta, \omega) = z^\eta(0, \omega).$$

The second part gives derivatives with respect to the control u and takes the form

$$\nabla_u \psi(\eta, \omega)(s) = h_u^\top(x^\eta(s, \omega), u(s), \omega) z^\eta(s, \omega),\ s \in [0, 1].$$

Both are expressed in terms of the solution $z^\eta(s, \omega)$ to the adjoint equation

$$\dot{z}(s, \omega) = -h_x^\top(x^\eta(s, \omega), u(s), \omega) z(s, \omega) \quad \text{for } s \in [0, 1), \tag{6.27}$$

$$z(1, \omega) = \nabla_x F(x^\eta(1, \omega), \omega). \tag{6.28}$$

The gradient $\nabla_\eta \psi(\cdot, \omega)$ is Lipschitz continuous and $\psi(\cdot, \omega)$ has a Frechet differential equal to $D\psi(\eta; \delta\eta; \omega)$ at $\eta \in \mathbf{H}$. These facts combined with the Dominated Convergence Theorem and Fubini's Theorem establish that J has a Frechet differential $DJ(\eta; \delta\eta)$ given by

$$DJ(\eta; \delta\eta) = \langle \nabla J(\eta), \delta\eta \rangle_{H_2}$$

with a Lipschitz continuous gradient of the form

$$\nabla J(\eta) = E\left[\nabla_\eta \psi(\eta, \omega)\right]. \tag{6.29}$$

Analogously to (6.26), we define an *optimality function* $\theta : \mathbf{H} \to (-\infty, 0]$ given by

$$\theta(\eta) = \min_{\eta' \in \mathbf{H}} DJ(\eta; \eta' - \eta) + \frac{1}{2}\|\eta' - \eta\|_{H_2}^2, \tag{6.30}$$

which is clearly nonpositive due to the fact that $DJ(\eta; \eta' - \eta) + \frac{1}{2}\|\eta' - \eta\|_{H_2}^2 = 0$ if $\eta' = \eta$. Moreover, one can show that θ is continuous and equals to zero at every local minimizer of Problem OCPU; see Phelps et al. (2016) for specific details and Polak (1997) and Royset and Wets (2015) for general treatments of optimality functions. Consequently, $\theta(\eta) = 0$ is an optimality condition for Problem OCPU. In the next section, we give an algorithm that generates a sequence of initial-conditions and control pairs $\{\eta^k = (\xi^k, u^k)\}_{k=1}^\infty$ with accumulation points that indeed satisfy this optimality condition.

6.3.3 Algorithm and Its Convergence

In Sect. 6.1.4, we alluded to a solution approach for Problem OCPU as given in (6.18) through the solution of the approximate problems (6.20). We now provide theoretical foundations for such an approach.

Although a naive implementation might involve the solution of a single approximate problem (6.20) for an appropriately large sample size N, there is ample empirical and some theoretical evidence that computing speeds can be improved by approximately solving a sequence of approximate problems with increasing N.

The main reason for this effect is that significant progress towards an optimal solution can be made by considering inexpensive approximate problems with low N and that the solution of refined but costly approximations with high N can be warm-started by an earlier solution. We refer to Pasupathy (2010), Royset (2013), and Royset and Szechtman (2013) for further discussions on the subject of gradually increasing the sample size.

To adopt an algorithmic approach based on gradually increasing the sample size and the approximate solution of the corresponding approximate problems, we need to have a manner of quantifying the level of accuracy in a solution of an approximate problem. We turn to optimality functions for this purpose, which can be derived analogously to the optimality function θ given above. Specifically, under mild assumptions we find that J^N has a Gateaux differential $DJ^N(\eta;\delta\eta)$ given by

$$DJ^N(\eta;\delta\eta) = \langle \nabla J^N(\eta), \delta\eta\rangle_{H_2},$$

with a Lipschitz continuous gradient given by

$$\nabla J^N(\eta) = \frac{1}{N}\sum_{i=1}^{N} \nabla_\eta \psi(\eta, \omega^i). \tag{6.31}$$

The optimality functions $\theta^N : \mathbf{H} \to (-\infty, 0]$ are then given by

$$\theta^N(\eta) = \min_{\eta'\in\mathbf{H}} DJ^N(\eta;\eta'-\eta) + \frac{1}{2}\|\eta'-\eta\|^2_{H_2}. \tag{6.32}$$

We can show that θ^N is continuous and equals to zero at every local minimizer of the approximate problem (6.20); see Phelps et al. (2016). Consequently, $\theta^N(\eta) = 0$ is an optimality condition for (6.20) and the value $\theta^N(\eta)$ quantifies in some sense the level of accuracy of η, with lower numbers indicating poorer accuracy. We are then ready to state the algorithm:

OCPU Algorithm.

Step 1. Select a sequence of integers $\mathcal{N} = \{N_1, N_2, \ldots\}$ such that $N_k \to \infty$ as $k \to \infty$, and tolerances $\{\epsilon^N\}_{N=1}^{\infty}$, with $\epsilon^N \geq 0$ for all N and $\epsilon^N \to 0$. Initiate the iteration counter by setting $k = 1$.
Step 2. Obtain an approximate solution η^k for (6.20) under sample size N_k that satisfies $\theta^{N_k}(\eta^k) \geq -\epsilon^{N_k}$.
Step 3. Replace k by $k+1$ and go to Step 2.

Step 2 of the algorithm can be carried out by well-developed standard optimal control solvers such as DIDO (see Elissar Global 2015). Of course, numerous implementation details remain for example related to choices of sample size, ϵ^N, and optimal control solver, but a discussion along those lines are beyond the scope of this text. Some indications are given in Sect. 6.2.

It is possible to establish that every accumulation point η of a sequence $\{\eta^k\}_{k=1}^N$ generated by the algorithm satisfies the optimality condition $\theta(\eta) = 0$. Consequently, even if (6.18) is never directly considered in the algorithm, convergence to a solution can still be achieved. A heuristic argument for this fact is that the increasing sample size (under independent sampling) ensures that $\limsup_{N\to\infty} \theta^N(\eta^N) \leq \theta(\eta)$ for any sequence $\{\eta^N\}_{N=1}^\infty$ converging to a point η. Moreover, the fact that $\epsilon^{N_k} \to 0$ as $k \to \infty$ implies that Step 2 of the algorithm drives $\theta^{N_k}(\eta^k)$ up to zero. Since also θ is a nonpositive function, we obtain that $\theta(\eta) = 0$. In addition, we know that every accumulation point of a sequence of globally optimal solutions of (6.20), under sample sizes $\{N_k\}_{k=1}^\infty$, must be a globally optimal solution of (6.18). Precise statements of these facts are given in Phelps et al. (2016).

6.4 Notes

The first attempt to consider a searcher whose motion is the solution of a controlled differential equation appears to be Lukka (1974). Initially, this was carried out in the context of a stationary target, but soon extended to conditional deterministic target motion (Lukka 1977a,b). Still, the searcher dynamics needed to belong to special classes and the focus was on necessary optimality conditions. The 1980s saw numerous extensions (Ohsumi 1984, 1986; Ohsumi and Mangel 1985; Sunahara et al. 1982a,b) especially in the development of sufficient conditions, which culminated with Ohsumi (1991); see Mangel (1988) for a summery of models and further references and Vereshchagin et al. (1980), Mangel (1981), Ohsumi (1989), and Ohsumi (1991) for early computational methods.

The models in Sect. 6.1 are based on Foraker (2011), Chung et al. (2011), and Foraker et al. (2015a). The numerical results in Sect. 6.2 are taken from Chung et al. (2011) and Phelps et al. (2016). Extensive numerical simulations are found in Foraker (2011), Phelps (2015), and Foraker et al. (2015b).

The consideration of searchers governed by nearly arbitrary dynamical systems was pioneered in Foraker (2011) (see also Foraker et al. 2015a) and Phelps et al. (2016), the latter serves as the basis for Sect. 6.3. The optimality conditions developed there follow the pioneering work of E. Polak on $\mathcal{L}_2$-variations and consistent approximations Polak (1997, Chapters 3 and 4), which naturally lead to implementable algorithms as illustrated in this chapter. A parallel development in the tradition of Pontryagin, also for general dynamical systems, is given in Phelps et al. (2014).

References

Chung H, Polak E, Royset JO, Sastry S (2011) On the optimal detection of an underwater intruder in a channel using unmanned underwater vehicles. Nav Res Logist 58(8):804–820

Elissar Global (2015) DIDO optimal control software. http://www.mathworks.com/products/connections/product_detail/product_61633.html

Foraker JC (2011) Optimal search for moving targets in continuous time and space using consistent approximations. Ph.D. thesis, Naval Postgraduate School, Monterey

Foraker JC, Royset JO, Kaminer I (2015a, to appear) Search-trajectory optimization: part 1, formulation and theory. J Optim Theory Appl p

Foraker JC, Royset JO, Kaminer I (2015b, to appear) Search-trajectory optimization: part 2, algorithms and computations. J Optim Theory Appl p

Gill P, Murray W, Saunders M (1998) User's guide for SNOPT 5.3: a Fortran package for large-scale nonlinear programming. Technical report, SOL-98-1, System Optimization Laboratory, Stanford University, Stanford

Lukka M (1974) On the optimal searching tracks for a stationary target. Technical report, University of Turku, Turku

Lukka M (1977a) On the optimal searching tracks for a moving target. SIAM J Appl Math 32(1):126–132

Lukka M (1977b) On the optimal searching tracks for a randomly moving target-a free terminal point problem. Technical report, Department of Mathematics, Lappeenranta University of Technology, Lappeenranta

Mangel M (1981) Search for a randomly moving object. SIAM J Appl Math 40(2):327–338. http://www.jstor.org/stable/2101007

Mangel M (1988) Search theory: a differential equations approach. In: Chudonovksy D, Chudonovsky G (eds) Search theory. Marcel Dekker, New York, pp 55–103

Ohsumi A (1984) Stochastic control with searching a randomly moving target. In: Proceedings 1984 American Control Conference, San Diego, pp 500–504

Ohsumi A (1986) Optimal searching for a markovian target and relation to optimal stochastic control. In: Byrnes C, Lindquist A (eds) Theory and applications of nonlinear control systems. North-Holland, Amsterdam, pp 569–583

Ohsumi A (1989) Algorithms for optimal searching and control systems for a Markovian target. In: Leondes C (ed) Control and dynamic systems, vol 30. Academic, New York, pp 99–118

Ohsumi A (1991) Optimal search for a Markovian target. Nav Res Logist 38(4):531–554

Ohsumi A, Mangel M (1985) Joint searching and parameter identification for a Markovian target. In: Proceedings 1985 American Control Conference, Boston, pp 517–525

Pasupathy R (2010) On choosing parameters in retrospective-approximation algorithms for stochastic root finding and simulation optimization. Oper Res 58(4):889–901

Phelps C (2015) Computational optimal control of nonlinear systems with parameter uncertainty. Ph.D. thesis, University of California, Santa Cruz

Phelps C, Gong Q, Royset J, Walton C, Kaminer I (2014) Consistent approximation of a nonlinear optimal control problem with uncertain parameters. Automatica 50(12):2987–2997

Phelps C, Royset JO, Gong Q (2016) Optimal control of uncertain systems using sample average approximations. SIAM J. Control and Optimization 54(1):1–29

Polak E (1997) Optimization. Algorithms and consistent approximations. Springer, New York

Ross IM, Karpenko M (2012) A review of pseudospectral optimal control: from theory to flight. Annu Rev Control 36:182–197

Royset JO (2013) On sample size control in sample average approximations for solving smooth stochastic programs. Comput Optim Appl 55(2):265–309

Royset JO, Szechtman R (2013) Optimal budget allocation for sample average approximation. Oper Res 61:762–776

Royset JO, Wets RJ (2015) Optimality functions and lopsided convergence. J Optim Theory Appl p (in review)

Sunahara Y, Ohsumi A, Kobayashi S (1982a) On a method for searching a randomly moving target. In: Proceedings of the 11th SICE symposium on control theory, Kobe

Sunahara Y, Ohsumi A, Kobayashi S (1982b) On the optimal search for a randomly moving target. In: Proceedings of the 5th SICE dynamical systems theory symposium, Yokosuka

Vereshchagin AF, Medvedev VS, Puzanov VP (1980) Algorithms for optimizing the trajectories of motion in the case of controllable search for objects. Eng Cybern 18:115–122

Wagner D, Mylander W, Sanders T (1999) Naval operations analysis. Naval Institute Press, Annapolis

Washburn A (2014) Search and detection, 5th edn. Create Space, North Charleston, SC

Washburn A (2014d) Search and detection, 5th edn. Create Space, North Charleston

Chapter 7
Search Games

So far in this book we have essentially assumed that the target is either indifferent to the outcome or unable to influence it. Any uncertainty about target location has been in accord with a stochastic model of some kind that the searcher understands. If there is uncertainty about initial location, as in Chap. 2, then a probability distribution for that location has been given. If there is also uncertainty about subsequent movement, as in later chapters, then a probability law for that motion has also been given. But the target of search might have an opinion about whether being found is actually a desirable outcome, and might also be able to influence it. In this chapter we will consider the target to be sentient, making his choices about location and movement to either discourage detection (Sect. 7.1) or encourage it (Sect. 7.2). The searcher will know the target's capabilities, but not his habits or intentions, so probability laws for the target's location and movement will be missing.

There is a large literature on this subject. Our goal is merely to introduce the main concepts and give some examples. See Alpern and Gal (2003) for a more comprehensive treatment and extensive references.

7.1 Uncooperative Targets

In this section we model search as a two-person zero-sum (TPZS) game where the searcher desires detection while the target wishes to avoid it. To emphasize the target's opposition he will be renamed as the "hider". There is no possibility for cooperation because the goals of the two decision makers are exactly opposite. Hide-and-Seek games played by children are an example. There are many examples involving military forces or terrorists where the outcome has more serious implications. A good reference for TPZS games in general is Washburn (2014a), which includes a chapter on search games.

L.D. Stone et al., *Optimal Search for Moving Targets*, International Series in Operations Research & Management Science 237,
DOI 10.1007/978-3-319-26899-6_7

7.1.1 TPZS Theory

The primary theoretical conclusion that we need is that, in games where each side has only a finite number of alternatives or "strategies" (the Game Theoretic term) to choose from, a solution will always exist if mixed strategies are permitted where the decision maker surrenders his choice to some kind of a randomization mechanism. To be precise, let the payoff when player 1 chooses strategy i and player 2 chooses strategy j be a_{ij}; all of these payoffs are assumed to be accurately known to both sides. We will always understand that the payoff is to player 1; that is, player 1 is the maximizing player and player 2 is the minimizing player. Then in an $m \times n$ game, there will always exist mixed strategies $\mathbf{x} = (x_1, \ldots, x_m)$ for player 1 and $\mathbf{y} = (y_1 \ldots, y_n)$ for player 2 and a number v such that

$$(1)\ \sum_{i=1}^{m} a_{ij} x_i \geq v \text{ for all } j, \text{ and } (2)\ \sum_{j=1}^{n} a_{ij} y_j \leq v \text{ for all } i \tag{7.1}$$

(see Sect. 17.6 of von Neumann and Morgenstern (1944)). The first set of inequalities states that the average payoff will be at least v no matter what player 2 does, as long as player 1 randomizes according to $\mathbf{x}$. The second set states that the average payoff will be at most v no matter what player 1 does, as long as player 2 randomizes according to $\mathbf{y}$. Together, these two statements establish v as the "value of the game" in the sense that either player can guarantee that average payoff, regardless of what the other player does. The two optimal mixed strategies $\mathbf{x}$ and $\mathbf{y}$ are in equilibrium in the sense that neither player, upon discovering the other's mixed strategy, has any positive motivation to change his own. It should be understood that the two randomizations are done essentially simultaneously; that is, neither player knows what the other player is going to do before making his own choice. That said, the optimal mixtures $\mathbf{x}$ and $\mathbf{y}$ do not themselves have to be kept secret or concealed from the adversaries.

The fundamental theorem (7.1) has been known since 1928 when von Neumann first proved it, but its truth is neither intuitive nor welcome for many human decision makers. A good illustration of this is the game "Rock-Paper-Scissors" where the outcome is either +1 (player 1 wins) −1 (player 2 wins) or 0 (a tie because both players have chosen the same strategy). With strategies numbered as in the name of the game, the payoff matrix is

$$\left(a_{ij}\right) = \begin{bmatrix} 0 & -1 & +1 \\ +1 & 0 & -1 \\ -1 & +1 & 0 \end{bmatrix} \tag{7.2}$$

It is easy to verify that both optimal mixed strategies are $(1, 1, 1)/3$; that is, each player should simply choose a strategy at random. If either player does so, the average payoff will be 0 regardless of what the other player does. A complete neophyte, armed with a die, can achieve a tie on the average against any other player, no matter how experienced. In other words, von Neumann has essentially spoiled the game by giving simple instructions for achieving its value against any other player. In spite of this observation, in 2016 there is still a society (www.worldrps.com) devoted to playing Rock-Paper-Scissors and even designating an annual champion. It seems to be a human fascination to try to predict the actions of others, even when randomization makes this impossible. Of course one could always prohibit the use of dice or other randomization devices in choosing strategies. Perhaps the World RPS society does that. We will not do so here, however, so von Neumann's theorem applies.

When the number of strategies for either player is infinite, there are examples of TPZS games that do not have solutions. A simple example is for each player to choose a number in the open interval (0,1), with the winner being the one who picks the larger number. That game has no solution because there is no largest number in that interval. All of the examples formulated below are solvable, however.

7.1.2 Games Without Movement

In this subsection the hider can choose his initial location, but must then stay there. Many such games have been solved, but only a few examples will be given here because this book is mainly concerned with targets that can move.

Our first example "Oneshot" has both parties simply choosing one of n cells. Suppose cell i has conditional detection probability p_i, and that detection happens with probability p_i if searcher and hider both choose cell i, or otherwise does not happen. For $n = 3$ the payoff matrix, which is as always known to both sides, is

$$\begin{bmatrix} p_1 & 0 & 0 \\ 0 & p_2 & 0 \\ 0 & 0 & p_3 \end{bmatrix} \tag{7.3}$$

The solution of this game has both players making the probability of choosing cell i be inversely proportional to p_i; that is

$$\mathbf{x} = \mathbf{y} = (c/p_1, c/p_2, c/p_3)\,, \tag{7.4}$$

where c is whatever constant is required to make the probabilities sum to 1. If $\mathbf{p} = (1, 0.5, 0.2)$, then $c = 1/7$ and the value of the game is also 1/7. The solution of even this simple game is somewhat surprising, since both sides are most likely to choose the cell with the smallest conditional detection probability. This makes sense for the hider, but may be counterintuitive for the searcher. Indeed, were we to formulate a search problem where the hider is equally likely to choose any cell, the best strategy for the searcher would be to emphasize the cell with the largest conditional detection probability, rather than the smallest. What we are calling an "optimal" mixed strategy for the searcher is sensitive to the assumption that the hider is aware that he is playing a game, and is therefore not equally likely to choose any cell. In search games, it is not unusual for the searcher to emphasize doing what he is worst at. Dear reader, if you are telling yourself that you would nonetheless choose the cell with the biggest detection probability (the first cell) if you were somehow forced to play this game, then consider the following facts:

1. always searching the first cell will always fail if the hider discovers your strategy,
2. using the optimal mixed strategy guarantees you 1/7, even if the hider discovers that you are using that strategy, and
3. the hider can guarantee a detection probability of at most 1/7 no matter what you do.

Search games where the searcher continuously distributes his time over the cells can also be formulated and sometimes solved, even though the number of strategies for player 1 is infinite. Consider a game that is the same as the search problem considered in Sect. 2.3.1.3, except that now the hider can choose any cell to hide in, rather than being constrained by a given distribution. Let $\mathbf{y} = (y_1, \ldots y_n)$ be the mixed strategy chosen by the hider, who is player 2 here because he wishes to minimize the detection probability. Also let $c(i) = 1$ and $b\,(i, z) = 1-\exp\,(-\alpha_i z)$ for all i; that is, we are considering a random search problem where all cells are equally costly to search, but where some cells are more difficult to search than others. If the searcher's allocation of search time to cells is $\mathbf{x} = (x_1, \ldots, x_n)$, we require that all components of $\mathbf{x}$ be nonnegative and $x_1 + \cdots + x_n \leq K$, where K is the total amount of time available for search. The probability of detection when $\mathbf{x}$ is matched against the hider's mixed strategy $\mathbf{y}$ is

$$A\,(\mathbf{x}, \mathbf{y}) \equiv \sum_{i=1}^{n} y_i\,(1 - \exp\,(-\alpha_i x_i)). \tag{7.5}$$

One attractive strategy for the searcher is to equalize the detection probability in all cells, since equalization takes away the hider's flexibility in choosing a cell. This idea leads to making $x_i = c/\alpha_i$, where $c = K/\,(1/\alpha_1 + \cdots + 1/\alpha_n)$, and to a lower bound on the game value of $v = 1 - \exp\,(-c)$. This lower bound turns out to be the value of the game; that is, equalization is optimal for the searcher and there is no need for him to consider mixed strategies. We have not yet found the optimal $\mathbf{y}$, but there nonetheless exists a $\mathbf{y}$ such that $A\,(\mathbf{x}, \mathbf{y}) \leq v$ for all feasible $\mathbf{x}$, even if the searcher knows $\mathbf{y}$. This game turns out to be a rare example of a game being easier to solve than the corresponding optimization problem—compare the simple expression for v above with the operations required to find the detection probability in Sect. 2.3.1.3.

Lest the reader conclude from the above two examples that optimal strategies can always be intuited, consider the following payoff matrix

$$\left(a_{ij}\right) = \begin{bmatrix} 0.3 & 0.1 & 0 \\ 0.2 & 0.8 & 0.3 \\ 0 & 0.1 & 0.5 \end{bmatrix} \tag{7.6}$$

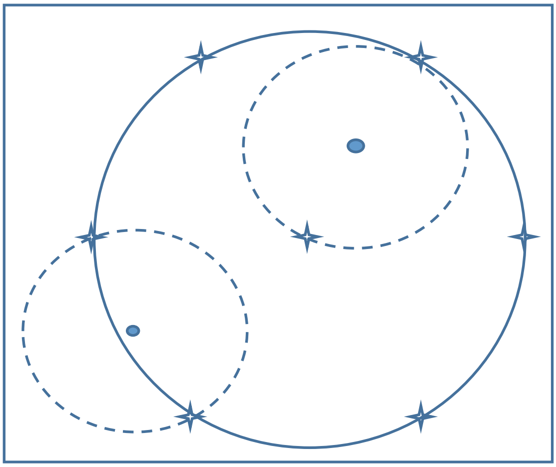
This might represent a situation where the searcher can detect the hider even if the searcher guesses the wrong cell, as long as the guess isn't too wrong. The general technique for solving games of this sort is Linear Programming. After applying that technique, we find that $\mathbf{x} = (0.25, 0.75, 0)$, $\mathbf{y} = (0.75, 0, 0.25)$ and $v = 0.225$. The searcher never searches in cell 3, and the hider never hides in cell 2. Neither of these characteristics is obvious *a priori*; indeed, it is often difficult to predict which strategies will be unused in a TPZS game. This difficulty led to serious doubts, prior to von Neumann's proof, about whether the fundamental theorem was actually true.

Search games need not be square. The hider might have his choice of five locations, for example, while the searcher has his choice of three sensor types. Linear Programming can be used to solve all of them.

The search space can also be continuous. A famous continuous example is the "Hiding in a disk" (HDG) game in Ruckle (1983). A detection radius d is given, and both parties select a point in the unit disk. Let z be the distance between the two points. The searcher wins if and only if $z < d$. We might expect the value of the game to be d^2, the ratio of the covered area to the area of the unit circle, but the value isn't quite that large because some of the searched area will often lie outside the unit circle. The exact value of the game is known only for certain values of d. The value is 1/7, for example (not 1/4) when $d = 1/2$. The hider can guarantee this by first selecting any six equally spaced points on the circumference, and then randomly choosing to hide at one of those six or at the center with probability 1/7 each. There is no way to include two of those points in the interior of a circle with radius 1/2 (see Fig. 7.1), so with only one look the searcher cannot do better than 1/7.

The complicating feature in HDG is edge effects. To avoid that problem, restrict both sides to the circumference of the unit circle, with d being the length (not the area) covered by the searcher. The value of that game is exactly d/π. One could also avoid edge effects by replacing the unit disk with the unit sphere.

Fig. 7.1 The HDG disk is shown as a *solid circle*. The *dashed circles* show two not-quite-successful attempts to include two of the seven stars in the interior of a circle with half the radius of the disk. It can't be done



7.1.3 *Games Extended in Time*

We now permit the hider to move from one place to another, perhaps in addition to being able to choose his initial location. Time is involved, so the hider selects a "track" rather than a "position". The searcher also generally moves from place to place, often having multiple chances at detection. The number of strategies available to the two sides explodes exponentially, as does the associated payoff matrix, so the usefulness of Linear Programming as a general purpose solution technique diminishes. Nonetheless, von Neumann's theorem still applies, and a variety of games have been considered and at least partially solved.

7.1.3.1 Unrestricted Motion

It is often the case that things simplify in the extremes, and search games are no exception. One extreme is where neither side can move, as in Sect. 7.1.2. The other extreme is where motion on both sides is completely unrestricted.

Consider NShot, which is simply Oneshot repeated N times with searcher and hider each able to select any cell at each time. At each time the hider can guarantee that the detection probability will not exceed v, the value of Oneshot, no matter where the searcher looks. By choosing his positions independently at each opportunity as in Oneshot, the hider can guarantee that the chance of at least one detection out of N tries will not exceed $1 - (1 - v)^N$. A similar argument for the searcher, who is also unrestricted, establishes that this quantity is the value of the game. The same argument works for any detection game that is repeated with both sides able to select a strategy independently on each play. If it is *possible* for both sides to select a strategy independently on each play, then it is also *optimal* for them to do so. However, independence may not be possible if the movement of at least one of the players is restricted between plays.

7.1.3.2 Restricted Searcher Movement

Isaacs (1965) introduced the Princess and Monster (P&M) game as a prototype for searching for a mobile hider in a restricted area. The frantic Princess is located within the unit disk, moving about while a blind Monster searches for her. The Monster's speed is limited to V, and he will detect the Princess if the distance between them ever gets to be as small as d, a capture distance much less than 1. The Princess is also blind, but can move as fast as she wants as long as she does not leave the unit disk, and can start anywhere within the unit disk. The payoff is the average amount of time for the Monster to detect the Princess, so the Princess is player 1, the maximizer. P&M generalizes Ruckle's HDG to allow movement by both players.

Even though both sides have infinitely many strategies, we might try to guess or at least bound the value of P&M. The Monster's sweep width is $2d$, so the rate at which he can cover area is at most $2Vd$. The area of the unit disk is π, so it will take the Monster $\tau \equiv \pi/(Vd)$ to cover the disk once. If he finds the Princess at the midway point, on the average, then the payoff would be $\tau/2$; in fact, the Princess could guarantee a time at least that large by simply hiding at a random point in the disk and never moving. But if she could spoil the Monster's attempted sanitization by moving about, thereby making his search random, rather than exhaustive, then the average time to detection would be twice as large (τ). But the Princess must be careful not to move too much, since finding the Monster is just as bad as vice versa. We are left wondering where the value of P&M is in the interval $[\tau/2, \tau]$, and what the Princess should do by way of movement.

Gal (1979) gives asymptotically optimal strategies for the two sides and an approximate value of τ for the P&M game. An optimal strategy for the Princess has her connecting a sequence of dots that are selected independently and uniformly in the disk. At each dot she pauses for a carefully selected amount of time before moving on to the next dot. Thus the Princess moves slowly, but still fast enough to introduce some unpredictable independence into her sequence of positions.

Lalley and Robbins (1988) give an appealing optimal strategy for the Monster. The strategy is a diffuse reflection wherein he constantly moves at top speed, reflecting randomly from the boundary of the region in the same manner as light reflects from a diffuse surface. That strategy has the property of equalizing the amount of time spent in all parts of the region, and is therefore an attractive candidate for motion that covers a region "randomly and uniformly" at constant speed. Regardless of how the Princess moves, this strategy guarantees a detection time of at most τ.

The net result of all this maneuvering by Princess and Monster is that the time to detection is approximately an exponential random variable with mean τ, which is what the Princess hoped to accomplish by moving. This is as close as game theory gets to making fairy tales come true. Capture is inevitable, unfortunately, so the Princess can only delay it as long as possible. If the P&M unit disk is replaced by any compact convex set with area A in two dimensions, then the value of the game is known to be $\tau \equiv A/(Vd)$, with the Monster's diffuse reflection strategy still being optimal.

Alpern and Gal (2003) describe many other search games played in bounded regions that involve a mobile hider and a mobile searcher with restricted speed. They all share the property that the game's value is inversely proportional to the searcher's speed and independent of the hider's speed, provided the hider's speed meets a certain lower threshold. The hider wants to move around just enough to prevent an exhaustive search—there is no hope of a complete escape when the region is bounded.

The above games all assume that the hider is just as ignorant of the searcher's position as vice versa, but there are real situations where the searcher's activity reveals his own position to the hider, who can sometimes make good use of the information. Imagine a discrete-time game based on four cells arranged cyclically in a square. Both players can choose any cell initially. The game ends when both choose the same cell, and the payoff is the time when the game ends. The hider has an advantage in that he can observe the location of the searcher after any unsuccessful look. If both players are free to choose any cell at each time, we have a repeated version of Oneshot whose value is 4—the searcher is so mobile that information about the previously searched cell is of no value to the hider. But suppose instead that the searcher is restricted to moving at most one cell in either direction, so that the cell diagonally opposite the most recent choice is not available for his next choice. The modified game will go on forever, since an undetected hider can always move to the cell diagonally opposite the searcher's most recent choice. The hider could do this even if his own movements were restricted like the searcher's. This is an extreme example—there is a great deal of difference between 4 and infinity—but it serves our purpose of illustrating the importance of information in a search game.

Actual solutions of search games where the hider has an information advantage are rare. For example consider the P&M game where the Princess at all times knows the direction to the Monster, call the game P&M+. Nobody knows its value, even approximately, except that it surely exceeds the value of P&M. This is unfortunate, since P&M+ resembles certain real-world situations where the searcher employs an active sensor. An active sensor will generally be detected by its target long before detecting it, thus revealing a direction to the sensor.

7.1.3.3 Restricted Hider Movement

A simple example would be to require the hider to start at the origin of the two-dimensional plane, maneuvering as he wishes over the interval [0,T] as long as his speed never exceeds U. At the final time T the searcher can search once at any point in the plane, succeeding if and only if the point that he chooses is within D of the hider at that time. Since the hider's position is irrelevant except at time T, and since he can clearly be anywhere within a circle of radius UT at time T, his strategy amounts to picking a point in the circle and then (it doesn't matter how) going there. In fact, this game is equivalent to HDG with $d = D/(UT)$. In this game the hider would clearly prefer a larger speed limit, or better yet to have no speed limit at all. However, there are also search games where the hider moves only because the rules force him to do so.

Suppose the hider is required to move from point A to point B, and that there are n routes from A to B, any one of which the searcher can interdict. If the searcher chooses route i and the hider also chooses that route, then the detection probability is p_i. This game amounts to Oneshot where the "cells" are routes. If the hider's mission did not require him to move, he could remain safe at A. More generally, the requirement to move makes the hider vulnerable to ambushes. Garnaev et. al. (1997) describe the solution of a generalized game where the hider moves from A to B over a network while the searcher makes repeated attempts to detect him.

While allowing both players complete freedom of motion results in simply playing the one-move game repeatedly, games where only one of the players has complete freedom can be surprisingly complex. Suppose there are only two cells to hide in, with p_1 and p_2 being the conditional detection probabilities. The searcher has complete freedom, but the hider must remain stationary once he chooses a cell. To make the game simple suppose $p_1 = 1$, and let the payoff be the expected number of looks required for detection. An attractive strategy for the searcher would be to look once in cell 1, and then (failing detection on the first look) look in cell 2 until detection occurs. The average number of looks required is 1 if the hider hides in cell 1, or $1 + 1/p_2$ if the hider hides in cell 2. However attractive, that strategy is not optimal for the searcher. If he followed it consistently then the hider would learn to hide in cell 2, and if the hider hides in cell 2, then the searcher should look in cell 2 first, rather than cell 1—there cannot be an equilibrium where the searcher always looks in cell 1 first. We can still solve the game by capitalizing on the fact that the hider has only two strategies. If $p_2 = 0.5$, for example, the game value is 2.4, with the searcher's only two active strategies being to look in cell 1 on the second or third look (construct a 2×2 payoff matrix to verify this statement). Our point here is that the game is more subtle than it might first appear. Now complicate the game by letting p_1 be arbitrary, or by considering more than two cells. Even enumerating the searcher's strategies becomes a formidable task, let alone solving the game. This game has not been solved in general. Bram (1963) describes what is known of the solution.

An exercise here would be to solve NShot with $\mathbf{p} = (1, 0.5)$. The reciprocal of the value of that game should exceed 2.4 because hider's motion is unrestricted in NShot. Does it?

Bram's game simplifies, of course, if all of the conditional detection probabilities are 1. Ruckle (1983) names this the Search on a Complete Graph (SCOM) game. With n cells there are n strategies for the hider and n-factorial strategies for the searcher, since he can search the cells in the order of any permutation. With 10 cells that would be 3,628,800 strategies, so writing out SCOM as a matrix game and applying Linear Programming is not attractive. Nonetheless the game is simple enough that we can intuit the solution. An optimal strategy for the hider is to pick a cell at random. As long as the searcher never searches a cell twice, this guarantees that the number of looks required is equally likely to be any number between 1 and n, or $(n + 1)/2$ on the average. It will take even longer if the searcher repeats a cell. An optimal mixed strategy for the searcher is to randomly choose any one of the permutations, but there are simpler mixed strategies that are also optimal. He can always choose the permutation $(1, \ldots, n)$, for example, as long as he is careful to start his search at a randomly selected point in the permutation and search the cells cyclically.

The same idea (always use the same permutation, but start at a random point within it) is also effective in a continuous game resembling SCOM that is played on a connected graph where the nodes are connected by arcs, the sum of the lengths of which is L, and where the hider can hide at any point on any arc. The searcher can start anywhere, but must then move over the arcs at speed V until he finds the hider. An Eulerian path is a sequence of arcs that eventually returns to the same point after covering every other point without duplication. Some graphs have such paths (a Fig. 8, for example), and some don't (most trees). As long as the graph has an Eulerian path, it is optimal for the searcher to pick a random point on it and move around the path until he finds the hider. The value of that game is $L/(2V)$, half the length of time for a complete circuit.

7.1.3.4 Restricted Searcher and Hider

Ruckle (1983) describes several multi-look games other than SCOM where there is no overlook probability and the hider cannot move. They are played on graphs, with the hider choosing a node of the graph and remaining there while the searcher moves over the arcs between looks. One of them is Search on a CYClic graph (SCYC) where the nodes are arranged on a circle and the searcher, after selecting any node for his initial search, can only move to one of the two neighbors of the node most recently searched. The searcher's cyclic optimal strategy in SCOM is still playable in SCYC, so the two games turn out to have the same solution. In fact we could even restrict the searcher to move clockwise without changing the value of the game—as long as we only forbid him to use strategies that he doesn't want to use anyway, he will not complain.

The SCYC game changes essentially if the hider's motion is restricted like the searcher's to only neighboring cells. Ruckle (1983) names this the Cyclic Pursuit Game (CPG) game, and has offered a reward to anyone who can solve it. So far the reward has gone unclaimed. Why should such an easily described game be so difficult to solve? The problem is that there are infinitely many strategies for both sides, and the game does not possess the kind of symmetry that makes it possible to guess the form of the optimal strategies. Consider the searcher strategy of employing his SCYC strategy in CPG. The hider can defeat this by playing the same strategy—even if the searcher flips a coin to decide which way to go, there is still a good chance that the searcher will follow the hider around the cycle indefinitely. Bounds for the game value have been obtained by having the players move in random walks, but the bounds are not equal.

Another famous game involving restricted motion is the Flaming Datum problem, named after a situation in WWII where a submarine has just torpedoed a merchant ship. The "flaming datum" marks a spot where the submarine once was when pursuing destroyers arrive sometime later in pursuit of the submarine. In the abstract we have a TPZS game with three parameters, namely the time late τ, the submarine's top speed U, and the aggregate sweep rate of the destroyers S. Typically S would be computed by multiplying the destroyer's speed by a sweepwidth, perhaps summing if multiple ships are involved, but the details are not important. How should the two parties maneuver, and what is the resulting probability of detection?

The Flaming Datum problem has never been solved, but we can at least approximate its value. First define the Farthest-on-circle (FOC) to be the gradually expanding circle that defines the limit of the hider's position. The area of the FOC at time t after the flaming datum is created is $A(t) \equiv \pi(Ut)^2$. If the searcher were able to constantly search randomly within the FOC, effectively scattering confetti at rate S, then the detection rate would be $\lambda(t) \equiv S/A(t)$ regardless of how the hider moves within the FOC. Here we use the word "rate" in the sense of a continuous time Markov process. If the hider were able to make his position always uniform within the FOC, as well as independent from time to time, he could guarantee the same thing regardless of how the searcher searches. In other words, those tactics for searcher and hider are in equilibrium, with the implied probability of detection for a search from τ to T being

$$p(T) = 1 - \exp\left(-\int_{\tau}^{T} \lambda(t)dt\right) = 1 - \exp\left(-\frac{S}{\pi U^2}\left(\frac{1}{\tau} - \frac{1}{T}\right)\right) \tag{7.7}$$

Note that this expression does not approach 1 as T approaches infinity—the FOC expands with the square of time while the area searched goes up only linearly with time, so continued search eventually becomes hopeless.

We seemingly have an unexpectedly simple solution to a game that is both complex and important. The only problem is that neither of the recommended tactics is feasible. Searching randomly is not feasible for a searcher whose track must be continuous, and similarly there is no way for a hider whose track must be continuous to make his position "independent from time to time". Even so, the same tactics are also infeasible in P&M, but are known to lead to the correct solution in that game. An additional argument in favor of formula (7.7) is Fig. 7.2 (Washburn (1978)), which shows the results of 295 replications of an experiment using Navy officers to play both searcher and hider. The theoretical prediction in that figure is formula (7.7) with $S = VW$. It fits the experimental cumulative distribution function rather well.

7.2 Cooperative Targets

In this section we assume that the lost target desires detection, perhaps even more so than the searcher, so we will cease referring to the target as a "hider". The problem is to achieve near coincidence of the positions of the two parties. The problem might better be described as one of "rendezvous" rather than "search", and we shall use that term at times. Seemingly this agreement of interest should simplify the problem of finding optimal tactics, but that is not the case. Much of the difficulty is traceable to vagueness in the concept of "being lost". After all, many a child has been found by his frantic parents, only to report that he doesn't understand all the fuss because he was never lost in the first place.

There are two points of view one can take about searching for a cooperative target, symmetric and asymmetric. The asymmetric case is the simpler of the two because it leads to a clear notion of who is the target and who is the searcher. This case is analyzed in Sect. 7.2.1. The more difficult symmetric case is taken up in Sect. 7.2.2.

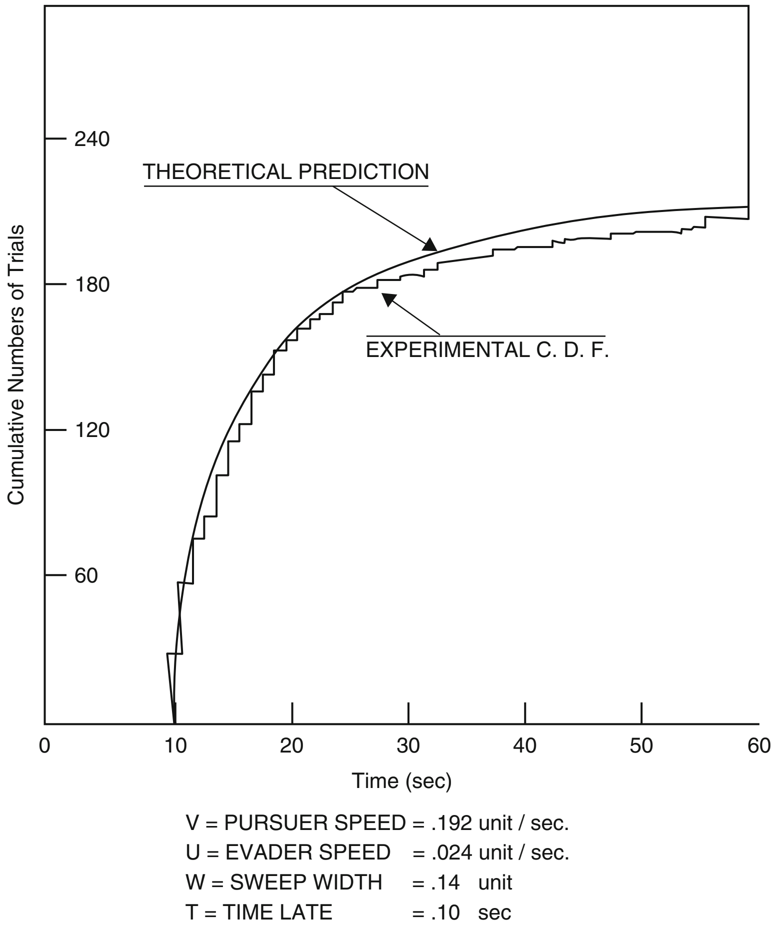


Fig. 7.2 Results of a Flaming Datum experiment

7.2.1 *The Asymmetric Case*

Here we consider problems where the roles of target and searcher are clear. The difficulty to be resolved is that the searcher does not know where the target is, even if the target does. We assume that the target is aware of the searcher's uncertainty and will help to resolve it by facilitating detection.

For the most part we will assume that the searcher's sensor is better than the target's, in which case a single detection distance d suffices, but suppose for a moment that the opposite is true. If the target first detects the searcher, then the question arises as to what the target can do to facilitate detection by the searcher. "Become more visible" is the obvious answer. Jump up and down, make noise or build a smoky fire, wave your arms, turn on your transponder, use a mirror to reflect the sun in the searcher's eyes, release a dye packet, etc. These tactics are so obvious that they do not require much thought or coordination. Target motion is another matter. The obvious tactic of moving directly toward the searcher (pursuit) is not necessarily the best one, especially if the searcher is making an exhaustive search. Exhaustive searches tend to be characterized by long straight segments where the searcher's motion is predictable, in which case the target would be better off to follow an intercept course, rather than a pursuit course, or possibly even move away from the searcher in order to be close to him on the next segment. The effect of following an intercept course is to effectively increase the searcher's sweep width, which is of course beneficial. The quantitative details about optimal tactics and the resulting increase in sweep width can be found in Washburn (2014b, pp. 11–13).

From here on we suppose that the searcher's sensors are superior, in which case a single detection distance d suffices. For an example of a typical problem, assume that the target is lost within some two-dimensional region S with area A, "lost" meaning that the target is equally likely to be anywhere. The search will continue until the distance between the two is smaller than d. The target can move, limited only by a top speed U, and similarly the searcher can move at top speed V. Let random variable T be the time required for detection. What should the two parties do to minimize $E(T)$?

The standard answer to this question is that the target should not move at all while the searcher conducts an exhaustive search of the region. This tactic for the target is employed so often that it has a name: "Wait for Mommy" or WFM. The sweep rate is $2Vd$, so detection will surely happen by time $\tau \equiv A/(2Vd)$, and $E(T)$ is half of that. The main virtue of WFM is that it permits an exhaustive search by the searcher. If U were larger than V it would of course be better for the "target" to conduct an exhaustive search while the searcher uses WFM, but we assume $V \geq U$. If the target were instead to move around at speed U, the likely effect would be to turn the searcher's efforts into a random search and thereby double the mean time to detection. It is true that motion by the target would result in some dynamic enhancement of the searcher's speed, but dynamic enhancement is a small effect compared to the effect of turning an exhaustive search into a random search.

WFM gets wide use in the real, two-dimensional world. Human children are sometimes explicitly instructed to use WFM in the event of becoming lost. Other species also employ WFM with youngsters, especially when the tactic has the additional advantage of making the youngster difficult for predators to find. Also, animals that are normally solitary occasionally have a need to find each other for mating. In such cases sometimes one sex will move around a lot while the other does not, which is sort of WFM. Characteristic of all of these situations is that the roles of searcher and target are clear.

WFM is not as attractive in one dimension. Suppose both parties are placed independently and uniformly at random in the unit interval, each with unit speed. One attractive rendezvous strategy would be for both to move left toward the origin, waiting there for the other if necessary. That would guarantee rendezvous, but why wait at the origin? If the target gets there first, he could shorten the rendezvous time by heading back toward the other end, shortly meeting the searcher who must have started out farther away from the origin. Of course the searcher should do the same thing – call this strategy "Reflect Left" or RL. It turns out that RL minimizes the mean time to rendezvous, although of course it is tied with "reflect right" (RR), which would work just as well. The mean time to rendezvous is 1/3, whereas it would be 2/3 without the reflection. In fact RL is optimal in more general circumstances. For one thing it happens to be symmetric, so it is also the solution to the symmetric rendezvous problem. Also the initial locations do not have to be uniform or even identical—as long as both initial positions are biased towards the origin, by which we mean specifically that the two density functions are each decreasing, use of RL (but not RR now) by both parties is the optimal joint strategy (RR is optimal if the locations are biased towards the other end).

The unit interval benefits from having two recognizable endpoints to reflect from, but what if the endpoints were joined together and not recognizable, so that the unit interval becomes a circle with unit circumference (imagine a tropical island where movement is possible only on the beach)? RL and RR are no longer feasible strategies, but it doesn't follow that WFM should replace one of them. With the target using WFM, the average time to detection will be 1/2, half of the time for the searcher to go all the way around the circle. If instead the target moves at top speed in the direction opposite to the searcher, the beneficial effect will be as if the searcher's speed is increased to 2, thus reducing the mean rendezvous time to 1/4. In this one-dimensional world, children might get instructed to always move clockwise when lost, while mommy remembers to always go counterclockwise. Of course much depends on getting the direction right—if the target accidentally moves in the same direction as the searcher, rendezvous might never happen! A target who does not know which direction is clockwise would be better off using WFM, rather than flipping a coin to decide which way to go. In fact WFM resurfaces as being optimal if there is no common notion of "clockwise", with mommy travelling around the circle until rendezvous.

7.2.2 The Symmetric Case

Here we assume symmetry between all the parties who are trying to rendezvous. In two dimensions imagine two parachutists trying to locate each other, or two astronauts separated on the surface of a small planet. The problem also can occur, and in fact frequently does occur in one dimension. Two backpackers sometimes become separated on a trail, with at least one of them being uncertain whether he is in front of or behind the other. What should they do to restore contact?

The constraining feature in the symmetric case is that both parties must follow the same policy, although variation can to some extent be achieved through the use of randomization. This unfortunate constraint could of course have been eliminated if only the parties had anticipated the possibility of becoming separated, but we assume that is not the case. The use of randomization turns out not only to be permitted, but (as we shall see) desired, which makes this an odd branch of analysis. Randomization is never needed in single-person decision problems. The previous chapters use the Theory of Probability on almost every page, but at no point does the searcher himself ever do anything random. The Random Search Formula owes its genesis to a skeptical attitude toward the efficiency of search, rather than to a deliberate attempt to search randomly. Randomization is also not needed in rendezvous problems when asymmetric strategies are permitted. However, we will find randomization to be an essential part of symmetric optimal strategies, although the reason here is different than is Sect. 7.1. In TPZS games the role of randomization is to introduce some unpredictability. In symmetric rendezvous problems, the role is to introduce some accidental asymmetry. Symmetric problems are in general more difficult than the equivalent asymmetric problems. For example, the symmetric solution of the circular rendezvous problem introduced at the end of Sect. 7.2.1 is still unknown—see Alpern and Gal (2003), Sect. 12.3 for a discussion.

We make only passing reference to one important symmetric strategy, that being to head for a notable landmark and just wait there until the other party does the same thing. That strategy depends on the existence of a unique landmark that cannot be lost by either party. In the rest of this subsection we will suppose that such landmarks do not exist. We will also assume that the two parties cannot create visible tracks on the landscape as they move, a simplifying feature that is often at odds with reality.

7.2.2.1 A Two-Dimensional Problem

Return to the same two-dimensional problem considered in Sect. 7.2.1, but this time symmetrically with $U = V$. The time required for an exhaustive search is τ. If we could just tell one of the parties to use WFM while the other searches exhaustively, the mean time to detection would be $\tau/2$, but we cannot do that because those tactics are not symmetric. One symmetric solution would be to have both parties search "exhaustively". Assume that the effect of that would be to turn the two exhaustive searches into a random search with an enhanced speed that is 27 % higher than V (see Koopman (1956) for justification of this percentage), so that the mean time to detection would be $\tau/1.27$. This time is disappointingly large, since we might have hoped that 1.27 would be closer to 2 than to 1, but we can possibly improve on it by using randomization.

The best randomized strategy is certainly not to tell each party to flip a coin to decide whether to use WFM or search exhaustively. That policy is symmetric, but the mean time to detection would include 25 % of infinity because both parties might choose WFM. However, consider the class of strategies where the probability of searching exhaustively in each period of length τ is p, and where the decision is repeated independently in each period, if necessary. The parameter p is to be adjusted to minimize the mean time to detection. There are three possible cases in each period. They are, with probabilities in [],

1. both parties choose WFM $[(1-p)^2]$
2. both parties search exhaustively $[p^2]$
3. one uses WFM while the other searches exhaustively $[2p\,(1-p)]$

Let μ be the unconditional mean time to detection. The simplest of the three cases is the first, since we know that detection will not happen in that period, so the additional time to detection will be $\tau + \mu$. That statement is easy to make, but its truth depends on recognizing that the situation after failure is identical to the original situation—we are employing the Renewal Theorem. We also know that detection will surely happen in case 3, the additional time required being $\tau/2$, on the average. This leaves only case 2.

In case 2, the additional time required is t, should detection happen at time t before τ, or else $\tau + \mu$ if detection does not happen in that period. Let $\lambda = 1.27/\tau$ be the detection rate, so that $f(t) = \lambda \exp(-\lambda t)$ is the density function of the time to detection. Then the additional time to detection in case 2 is

$$\int_0^\tau tf(t)dt + \int_\tau^\infty (\tau + \mu) f(t)dt = \tau \frac{1 - (1 + \lambda\tau)\exp(-\lambda\tau)}{\lambda\tau} + (\tau + \mu)\exp(-\lambda\tau)\,. \tag{7.8}$$

Since $\lambda\tau = 1.27$, (7.8) is equivalent to $0.567\tau + 0.281\mu$. Taking account of all three cases, we have

$$\mu = (1-p)^2(\tau + \mu) + p^2(\tau/2) + 2p(1-p)(0.567\tau + 0.281\mu)\,. \tag{7.9}$$

We can solve (7.9) for μ and then adjust p to make μ as small as possible. The best value for p turns out to be 0.82, at which point $\mu = \tau/1.39$. The mean rendezvous time in practice might be larger because we have not dealt with the possibility that the two parties will have different estimates of τ or the origin of time. Putting aside those reservations, we have an upper bound of $\tau/1.39$ on the minimal rendezvous time. This is disappointing, since 1.39 is still rather far from 2. A truly optimal symmetric strategy, if one exists, is unknown. It should be clear that a high price is being paid for symmetry. The two parties would be much better off if the roles of searcher and target had been established before loss occurred.

Given that the optimal symmetric strategy will usually involve randomization, it should come as no surprise that only a few rendezvous problems have actually been solved in the sense that the minimizing symmetric strategy is known. One of them—rendezvous on the unit interval—was mentioned in Sect. 7.2.1 A discrete counterpart would have two labeled, recognizable cells, say "Left" and "Right", with each party occupying one of them and able to move to the other cell in unit time. The best symmetric strategy is "stay where you are in the Left cell, or otherwise move to the Left cell", or of course Left could be replaced by Right in describing that strategy. If the two parties start out in different cells, the rendezvous time with that symmetric strategy is one time unit, which is clearly minimal. The situation changes if the cells are not recognizable, but the best rendezvous strategy is still known. It is "flip a coin at each time to decide whether to move or stay put". Rendezvous will occur in two time units, on the average, and it is not possible to improve on that figure. Note that the ends of the unit interval were implicitly assumed to be labeled in Sect. 7.2.1; the rendezvous problem would be significantly different if one of the parties did not have a reliable sense of direction.

7.2.2.2 Rendezvous Problems on the Whole Line

Here we consider problems where both parties are initially located on the line, and the problem is to achieve coincidence. For a simple case assume that $U=0$ and $V=1$. Since the target's speed is 0, the fact that he desires detection is irrelevant and we have a single-person optimization problem where randomized tactics are not needed. Given a density function for the location of the target, how should the searcher move in order to minimize the expected time to detection? This problem is called the Linear Search Problem (LSP), and it has a long history starting with Bellman (1963), who introduced it. The best tactic for the searcher is always a zigzag path with increasingly long legs, so the problem amounts to determination of the optimal leg lengths. If the initial distribution of the target's position is standard unit normal, then the minimal time to rendezvous is known to be either 2.16 if the searcher can choose his starting point, or 2.90 if the searcher is forced to start at the origin. In the former case the searcher's best starting point is -1.57 (rather far out on the flank of the unit normal), after which he next proceeds to 2.74 before turning back towards the origin (Washburn (1995)). In fact he will probably not have to turn back towards the origin, since his chances of finding the target on his first leg are 0.94.

Having disposed of the case where $U = 0$, consider next the symmetric case where both parties have unit speed, and to make things really simple suppose that the initial distance between the two parties is for some reason known to be exactly 2. This problem would be trivial if the two parties knew who was on the left and who was on the right (the detection time would be 1 after each moves towards the other), but they do not. It would also be simple if we could use asymmetric strategies. We could advise the target to use WFM and the searcher to first move 2 units one way and then 4 units the other way. The mean time to detection would be 4, the average of 2 and 6. However, only symmetric strategies are permitted, and the parties do not know who is on the left and who is on the right.

One symmetric strategy would be for each party to pick a random direction and then move one step forward followed by one step backward, repeated as often as necessary. This strategy is distance-preserving in the sense that, if rendezvous does not happen, the situation will be exactly as at the beginning, with the two parties being separated by 2. Rendezvous will happen only if the party on the right goes left while the party on the left goes right, which happens with probability ¼, or otherwise the situation will not have changed. Therefore the mean time to detection satisfies the renewal equation

$$\mu = (1/4)\,1 + (3/4)\,(2 + \mu)\,, \tag{7.10}$$

the solution of which is $\mu = 7$. A better distance-preserving strategy is to move one step forward and two steps backwards, which can be shown to have a mean time to detection of 5, a significant improvement but still not as small as 4. The best known symmetric strategy is considerably more complicated and has a mean time to detection of 4.40 (Baston 1999). All of these strategies are distance-preserving, a large analytic advantage because it enables application of the Renewal Theorem as in (7.10). However, it is not known whether the best strategy is in fact distance-preserving, or even what the best distance-preserving strategy is.

If our plan was to first quickly dispose of the problem where the initial distance is exactly 2 before moving on to problems where the initial distance distribution is more general, then the plan has not worked out. Even the problem where the initial distance is known exactly turns out to be difficult when only symmetric strategies are permitted (in fact even the asymmetric problem is difficult; Alpern and Beck (1999)). The problem of finding optimal symmetric strategies seems to be difficult in general, and there are indications that the optimal strategies, even if they eventually become known, will involve randomization and be so complicated that at least one of the parties involved would in the real world be incompetent for performance. This is unfortunate, since rendezvous problems do occur in the real world.

The reader deserves at least something practical for having read all this way, so here is some advice if you happen to be a backpacker. The situation is that your partner hikes faster than you do, so he has preceded you on the trail with a promise to wait for you somewhere. Since parting, you have hiked for an hour without seeing him, and have begun to suspect that he might actually be behind you. Perhaps he left the trail for some reason, and you passed him there, or perhaps he took a wrong turn and has yet to realize it. You have the food and he has the tent, so both of you are going to be miserable if you don't find each other before nightfall. Like most backpackers, you have not previously discussed this situation with your partner. What should you do? The answer is PUT DOWN YOUR PACK! Don't hide it behind a tree, but put it right beside the trail where your partner will surely see it if he passes that way. If you can, make a directional arrow on the ground before pursuing your search, planning to periodically return to your pack and reverse the arrow as you explore the other direction. You will be better at searching without having a pack to carry, and the pack will play WFM while you are away from it. If you find your partner's pack, then move it to the other side of the trail, thus revealing to him that you have been there while you go to fetch your own pack. If you find that your own pack has been moved, then rejoice because the problem is nearly over. That strategy is symmetric, so, if we can just get everybody to read this book, you can count on your partner doing the same thing.

References

Alpern S, Beck A (1999) Asymmetric rendezvous on the line is a double linear search problem. Math Oper Res 24(3):604–618

Alpern S, Gal S (2003) The theory of search games and rendezvous. Kluwer, Boston

Baston V (1999) Two rendezvous search problems on the line. Nav Res Logist 46:335–340

Bellman R (1963) Research problem no. 63–9. SIAM Rev 5:274
Bram J (1963) A 2-player N-region search game. Operations evaluation group report OEG IRM-31 (AD 402914), Chief of Naval Operations, Washington, DC
Gal S (1979) Search games with mobile and immobile hider. SIAM J Control Optim 17(1):99–122
Garnaev A, Garnaeva G, Goutal P (1997) On the infiltration game. Int J Game theory 26:215–221
Isaacs R (1965) Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization. John Wiley & Sons, New York
Koopman B (1956) The theory of search, Part I. Kinematic bases. Oper Res 4:324–346
Lalley S, Robbins H (1988) Stochastic search in a convex region. Probab Theory Relat Fields 77(1):99–116
Ruckle W (1983) Geometric games and their applications. Pitman, Boston
von Neumann J, Morgenstern O (1944) Theory of games and economic behavior. Princeton U. Press, Princeton
Washburn A (1978) Results of playing an evasion game. Technical report NPS55Ws783, Naval Postgraduate School, Monterey
Washburn A (1995) Dynamic programming and the Backpacker's linear search problem. J Comput Appl Math 60:357–365
Washburn A (2014a) Two-person zero-sum games, 4th edn. Springer, New York
Washburn A (2014b) Search and detection, 5th edn. Create Space, North Charleston, SC
Washburn A (2014c) Search and detection, 5th edn. Create Space, North Charleston

Index

L.D. Stone et al., *Optimal Search for Moving Targets*, International Series in Operations Research & Management Science 237,
DOI 10.1007/978-3-319-26899-6

S

T

U

W