Toward Real-Time Planning for Robotic Search

Harun Yetkin

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Daniel J. Stilwell, Chair
William T. Baumann
A. A. (Louis) Beex
Pratap Tokekar
Hongxiao Zhu

July 20, 2018
Blacksburg, Virginia

Keywords: search theory, path planning, environmental characterization, MCTS,
multi-agent search, when to communicate

Toward Real-Time Planning for Robotic Search

Harun Yetkin

(ABSTRACT)

This work addresses applications of search theory where a mobile search agent seeks to find an unknown number of stationary targets randomly distributed in a bounded search domain. We assume that the search mission is subject to a time or distance constraint, and that the local environmental conditions affect sensor performance. Because the environment varies by location, the effectiveness of the search sensor also varies by location. Our contribution to search theory includes new decision-theoretic approaches for generating optimal search plans in the presence of false alarms and uncertain environmental variability. We also formally define the value of environmental information for improving the effectiveness of a search mission, and we develop methods for optimal deployment of assets that can acquire environmental information in order to improve search effectiveness. Finally, we extend our research to the case of multiple cooperating search agents. For the case that inter-agent communication is severely bandwidth-limited, such as in subsea applications, we propose a method for assessing the expected value of inter-agent communication relative to joint search effectiveness. Our results lead to a method for determining when search agents should communicate. Our contributions to search theory address important applications that range from subsea mine-hunting to post-disaster search and rescue applications.

Toward Real-Time Planning for Robotic Search

Harun Yetkin

(GENERAL AUDIENCE ABSTRACT)

We address search applications where a mobile search agent seeks to find an unknown number of stationary targets randomly distributed in a bounded search domain. The search agent is equipped with a search sensor that detects the targets at a location. Sensor measurements are often imperfect due to possible missed detections and false alarms. We also consider that the local environmental conditions affect the quality of the data acquired from the search sensor. For instance, if we are searching for a target that has a rocky shape, we expect that it will be harder to find that target in a rocky environment. We consider that the search mission is subject to a time or distance constraint, and thus, search can be performed on only a subset of locations. Our goal in this study is to formally determine where to acquire the search measurements so that the search effectiveness can be maximized.

We also formally define the value of acquiring environmental information for improving the effectiveness of a search mission, and we develop methods for optimal deployment of assets that can acquire environmental information in order to improve search effectiveness. Finally, we address the cases where multiple search assets collaboratively search the environment and they can communicate their local information with each other. We are particularly interested in determining when a vehicle should communicate with another vehicle so that the joint search effectiveness can be improved. Our contributions to search theory address important applications that range from subsea mine-hunting to post-disaster search and rescue applications.

Dedication

To Mom. Dokunduğu herşeyi güzelleştiren anneme.

Acknowledgments

A PhD is a long and challenging journey specifically when your spouse is also a graudate student and you are a parent of a joyful toddler whose favorite time is the time that you spend all together. Other than the family, I received the most help during this challenging journey from my advisor Dr. Daniel Stilwell. His positive encouragement and his mentorship throughout this journey helped me to see the finish line. He set an example of the academical and ethical excellence. I would like to sincerely thank him for his guidance and his help.

I would like to thank my committee members Dr. A. Louis Beex, William Baumann, Pratap Tokekar and Hongxiao Zhu for their support and encouragement. The extensive feedback on the written manuscript provided by Dr. Beex greatly improved the quality of this dissertation. The fruitful discussions with the committee members, specifically with Dr. Tokekar, opened new research questions that I am determined to solve in the future. I would also like to thank our collaborators James McMahon and Artur Wolek from Naval Research Laboratory at Washington DC for giving us the opportunity to transfer our results to Naval search applications.

I would like to thank to past and present members of Autonomous Systems and Control Laboratory at Virginia Tech. Many thanks to all my friends in Blacksburg, specifically to Bilgiday, Sinan, Serdar and many others for their great friendship and for the memorable time that we spent together. They have been the family in Blacksburg and they will remain so.

Finally, I would like to thank to my family for so many things. My mom, Emine Dolek, and my mother-in-law, Zeliha Alisan, spent tremendous effort to help us with taking care of our daughter. Without them, it would not be possible to complete the PhD. There is no word to explain how grateful my wife and I are for having such great parents. I am

deeply grateful to Abdullatif Cindoglu, Tarik Erbolat and Ramazan Lacin for believing in me and for their unconditional support to initiate my journey. Also, many thanks to my brother Hakan Ali and my sister Behiye, and to my brother-in-laws Niyazi and Erkan for their support and encouragement. Most of all, I am thankful for my daughter, Duru, for the joy and happiness she gave to us, and I am thankful for my wife, Aylin Alisan Yetkin, for helping me overcome every obstacle both in my academic life and in my personal life. She has been an amazing wife, an amazing mother, and an amazing friend for all these years. I am excited for the next journey that we will be taking together.

This research was funded by ONR grants N00014-12-1-0055 and N00014-16-1-2092. In addition, during my PhD study, I received financial support from the Bradley Department of Electrical and Computer Engineering at Virginia Tech for two semesters and from the Ministry of National Education in Turkey for one semester. I am very grateful to them for giving me this opportunity.

Contents

List of Figures x

List of Tables xiii

1 Introduction 1

1.1 Motivation . . . 2

1.2 Related work . . . 4

1.3 Content of this dissertation . . . 9

1.3.1 Single-agent search problem . . . 9

1.3.2 Characterizing the local environmental conditions . . . 11

1.3.3 Multi-agent search problem . . . 12

1.4 Dissertation overview . . . 13

2 Computing Optimal Search Paths 15

2.1 Preliminaries . . . 15

2.1.1 Problem formulation . . . 15

2.1.2 Search sensor model . . . 16

2.1.3 Belief update rule for the search vehicle . . . 18

2.2 Search objectives . . . 18

2.2.1 Objective 1: maximize estimation accuracy . . . 19

2.2.2 Objective 2: minimize the penalty due to incorrect estimations . . . 22

2.3 Notes on the search environment and the search vehicle . . . 24

2.3.1 Search environment . . . 24

2.3.2 Search vehicle . . . 25
2.4 Numerical results . . . 26
2.4.1 Effect of environmental uncertainty on search performance . . . 29

3 Search and Environment Characterization 32
3.1 Related work . . . 33
3.2 Preliminaries . . . 33
3.2.1 Environment sensor model . . . 34
3.2.2 Belief update rule for the environment vehicle . . . 34
3.2.3 Belief update rule for the search/environmental characterization vehicle 35
3.3 Sensors operate on separate vehicles . . . 35
3.3.1 Entropy change maximization . . . 36
3.3.2 Environmental loss function . . . 37
3.3.3 Path planning for the case characterization precedes search . . . 39
3.3.4 Path planning when both vehicles operate simultaneously . . . 41
3.4 Sensors operate on the same vehicle . . . 42
3.4.1 Search objective: maximize estimation accuracy . . . 42
3.4.2 Search objective: minimize the risk associated with incorrect estimations 44
3.5 Computing near-optimal paths for environmental characterization vehicle . . 45
3.5.1 Approximating the characterization gain of a path . . . 45
3.5.2 Approximating the characterization gain of a cell . . . 51
3.6 Numerical results . . . 53
3.6.1 Both sensors operating simultaneously on a single vehicle . . . 57
3.6.2 Each sensor on separate vehicles . . . 59

4 Switching Between Search and Characterization 62
4.1 Related work . . . 62
4.2 Achieving a target level of risk reduction . . . 65
4.2.1 Probability distribution on risk reduction . . . 66
4.2.2 Gain of selecting a sequence of actions . . . 67
4.2.3 Reducing computational complexity of the solution . . . 69

4.3 Numerical results 70

5 Multi-Vehicle Search Problem **76**

5.1 Background 76

5.2 Related work 78

5.3 Multi-vehicle search problem 81

5.3.1 Computing optimal paths in plan-time 83

5.3.2 Computing optimal communication actions in run-time 85

5.4 Numerical results 91

6 Approximate Solutions for Search Problems **95**

6.1 Approximation algorithms 96

6.1.1 Branch-and-bound method 96

6.1.2 Monte Carlo tree search 98

6.2 Applying branch-and-bound approach to search problem 100

6.2.1 Exact branch-and-bound planner 101

6.2.2 Approximate branch-and-bound planner 101

6.2.3 Numerical results 103

6.3 Applying Monte Carlo tree search approach to search problem 107

6.3.1 Vehicle model and action costs 108

6.3.2 Random-lines heuristic policy 109

6.3.3 Voxelized state heuristic policy 112

6.3.4 Numerical results 113

7 Conclusions and Future Work **118**

Bibliography **120**

Appendix A Bayes Estimators for Specific Value Functions **134**

List of Figures

2.1 Search area and cell-wise environment distributions 26
2.2 Results for mowing-the-lawn search and adaptive search for different path lengths. (a) planned path with mowing-the-lawn search (b) and adaptive search when path length is 7 (c) path length is $15, 50$ (d) path length is 25. 28
2.3 Frequency of estimation accuracy for mowing-the-lawn (MTL) and adaptive search with receding horizon strategy (RHC) for different path lengths . . . 29
2.4 Frequency of deviation from actual expected estimation accuracy for different ranges of μ. (a) when $16 \leq \mu \leq 18$, (b) when $10 \leq \mu \leq 12$, (c) when $3 \leq \mu \leq 5$ 30

3.1 Search area and cell-wise environment distributions 54
3.2 Optimal trajectories for search and characterization. Figures (a-c): trajectories for the case both sensors operate on the same vehicle when (a) proposed approach is employed (b) entropy change maximization method is employed, and (c) the mowing-the-lawn approach is employed. Figures (e-f): characterization vehicle's trajectories for the case the sensors operate on separate vehicles when the characterization locations are selected (e) by our proposed approach, (f) by entropy change maximization method. Figure (d) shows the search vehicle's trajectory when no environment information is acquired. . . 55

3.3 Percentage of occurrences for (a) error in search performance and (b) actual search performance when both sensors operate on *the same vehicle*. From top to bottom, (a.1) and (b.1) correspond to the proposed approach, (a.2) and (b.2) correspond to the entropy change maximization method, (a.3) and (b.3) correspond to the mowing-the-lawn approach, and (a.4) and (b.4) correspond to the case where environment information is not available. Note that the horizontal axis is the negative log of the results. Smaller values for (a) imply less error in search performance and larger values for (b) imply better search performance. 58

3.4 Percentage of occurrences for (a) error in search performance and (b) actual search performance when search and characterization are performed on *separate vehicles*. From top to bottom, (a.1) and (b.1) correspond to the proposed approach, (a.2) and (b.2) correspond to the entropy change maximization method, and (a.3) and (b.3) correspond to the case where environment information is not available. Note that the horizontal axis is the negative log of the results. Smaller values for (a) imply less error in search performance and larger values for (b) imply better search performance. 60

4.1 Search area and cell-wise environment distributions 72

4.2 Best path and best sequence of actions when (a) $\beta = 13$ and $\mathcal{B} = 0.85$, (b) $\beta = 13$ and $\mathcal{B} = 0.85$ and $z_1 = z_2 = 2$ are observed, (c) $\beta = 11$ and $\mathcal{B} = 0.85$, and (d) $\beta = 23$ and $\mathcal{B} = 0.65$. 74

5.1 Alternative paths for vehicle 1 to communicate with vehicle 2. a) τ_0 is the initial node to reason about communication. Dashed red line is the future trajectory for vehicle 2. Starting at node τ_0, vehicle 1 can take one of the alternative paths b) when taking the shortest path, communication occurs at node τ_5, c) otherwise, it can either occur at node τ_9 or d) at node τ_{10}. . . . 86

5.2 Search area and cell-wise environment distributions 92

5.3 A scenario where communication improves joint performance. Figures show a) the initial paths for both vehicles, and b) updated paths after communication occurs. 93
5.4 Comparisons of a) attained joint risk reduction after a mission, and b) the number of times the value of communication is computed 94

6.1 Search area and cell-wise environment distributions 104
6.2 Search paths for a) exact-solution method b) approximate solution method when mission length is 50 and c) exact-solution method d) approximate solution method when mission length is 100. 105
6.3 Comparison of average computation time (a) and average normalized search performance (b) between exact and approximate algorithms based on Monte Carlo simulation. Error bars indicate one standard deviation. The figure is created by Artur Wolek 106
6.4 A subsea search environment showing an unpublished data set from the Boston Harbor taken in 2016. The x and y axis represent unit-less dimensions corresponding to the number of cells in the image (61×61). Cell-wise rewards are normalized to 1. The figure is created by James McMahon 114
6.5 Parameter study of the (a) RANDOM-LINES and (b) VOXEL heuristics for a mission length of 1000. Each data point represents the reward averaged over 8 simulations. The figure is created by James McMahon 115
6.6 Attained reward of the various planners as a function of the mission length.The lines represent the mean value and the standard deviation (shaded regions) over the set of experiments. The figure is created by James McMahon 116
6.7 Traversed paths for a mission length of 1000 when RANDOM-LINES heuristic is employed (a), when VOXEL heuristic is employed (b), and when a random walk heuristic is employed. The trajectories begin at blue (bottom left of map) and end at yellow. The figure is created by James McMahon 116

List of Tables

2.1 Environment types 27
2.2 Search performance 28

4.1 Attained risk reduction with deterministic environments 71

Chapter 1

Introduction

We address search applications where a mobile search agent is tasked with finding an unknown number of targets in a bounded search area, and the search mission is subject to a time or distance constraint. It is assumed that the agent is equipped with a sensor to detect the targets and the performance of the sensor is affected by the local environmental conditions. It is also assumed that the environment at a location may not be known deterministically, but a probabilistic knowledge on the environmental conditions is provided to the agent. We present a decision-theoretic approach to compute the optimal search locations such that the probability of correctly estimating the number of targets at a location is maximized. Our decision-theoretic approach accounts for false negatives, false positives, and uncertainty in the environment. To the best of our knowledge, this is the first study to offer a solution to the search problem where all of these factors - multiple targets, existence of false positives, and uncertainty in the environment - are addressed.

When there is also an environment sensor that detects the local environmental conditions at a location, the question is where to optimally sense the environment so that the performance of a search mission can be improved. We consider several cases where the search sensor and the environment sensor are either placed on the same agent, or they are placed on separate agents. For each case, we present a formal method of computing the optimal locations where environment measurements should be acquired.

Finally, we extend our results for the single-agent search problem to the multi-agent case, where a number of search agents are assumed to collaboratively search the area to maximize

a joint performance. We consider that the agents can communicate with each other when they are sufficiently close. We are particularly interested in optimally determining when an agent should communicate with another agent.

In Section 1.1, we provide a few motivational examples to illustrate how the findings of this study can apply to real-world search applications. In Section 1.2, we present an extensive literature review of search theory. In Section 1.3, we introduce the specific problems considered in this dissertation and we briefly state our approach for each problem. We provide an overview of the dissertation in Section 1.4.

1.1 Motivation

Search theory has its roots in numerous civilian and military applications. The ground work of the theory was established through Naval search applications during the Second World War, and the theory subsequently received attention from the operations research community. The results of the theory also addressed many non-military applications. To name a few, search theory results are often used in exploring for oil deposits [1,2], searching for a lost party [3–5], and medical screening [6]. A particularly interesting application area of search theory is the estimation of fish population and finding fish stocks [7,8]. In this dissertation, we address search applications where a mobile search agent is to find an unknown number of stationary targets randomly distributed in a discrete and finite search environment and the search mission is subject to a time/distance constraint. Our results inform search applications such as mine-hunting missions and search and rescue operations.

In a mine-hunting mission, the goal is to locate the subsea targets (mines) in a bounded domain so that in a follow-on mission the found mines can be neutralized. The search mission is subject to a time or distance constraint. The practical interpretation of this constraint could be the limited battery capacity or the presence of a time window to perform the mission. Prior knowledge on the distribution of the targets may or may not be available, and the search agent seeks to maximize the probability that its estimate of the number of targets is as close as possible to the true number of targets in the search environment. The performance of the search mission can be dependent on the environmental conditions; mine-

hunting applications, for example, are usually conducted in shallow water and coastal areas where environmental factors such as sediment type strongly determine the effectiveness of sensor performance [9]. The question is where to acquire the measurements to improve the overall search performance given a time/distance constraint on the mission.

Similarly, in search and rescue operations, the goal is often to locate, give medical treatment, and extricate the victims/survivors. In most cases, medical treatment should be provided as early as possible. Thus, the task of searching for the survivors has to be performed within a limited time. Fires, fog, cluttered spaces and many other factors can make it harder to detect the survivors in some locations than in others. An optimal use of search effort can help to rescue more survivors in need.

Suppose that in addition to the search sensor, we also have an environment characterization sensor that detects the local environmental conditions. That is, after sampling a location with the environment characterization sensor, we acquire noisy measurements of the environment at that location and reduce the uncertainty about the environment. Because search performance is dependent on the environment, precise knowledge of the environment can improve search performance due to better search plans. For example, in a mine-hunting mission, one may choose to avoid searching areas that are known to contain excessive clutter and many false positives in favor of environments with few false positives. Similarly, in a search and rescue operation, one may choose to avoid areas with limited visibility in favor of open spaces with increased visibility. In situations where the environment is poorly known, effort to acquire environmental information may lead to improved search effectiveness.

In search and rescue operations, using multiple search agents to explore the search environment can significantly speed up the process of localizing the survivors, which is crucial in increasing the overall effectiveness of the operation. A fundamental question in multi-agent search problems is to determine when each agent should share its local information with the other agents. Considering that in some search applications communication is carried out through ultra-low bandwidth channels and hence extremely expensive, determining when a communication action should occur is a notoriously difficult problem. For instance, underwater acoustic communication is severely limited and restricted to a short communication range [10]. Thus, it is very challenging to design optimal path planning strategies for a team

of search agents mapping the ocean sea floor or performing a mine-hunting mission. Efficient techniques must be developed for collaborative search problems with limited communication.

1.2 Related work

Search theory is mainly concerned with finding an optimal allocation of available search effort to locate a lost or hidden target, such that a reward specified as a measure of search effectiveness is maximized. Bernard Koopman, a former researcher in the U.S. Anti-Submarine Warfare Operations Research Group, is the first to offer a systematic approach to the search problem [11]. Early works in search theory focused on the discovery of a single target known to be somewhere in a given continuous region or in a discrete set of possible locations. The sensing device attached to the search agent was assumed to acquire noisy observations of whether the target was present or not. Noisy sensor measurements included the false negatives, i.e. failing to detect a target that is present, in most of the early works. However, the issue of false positives, i.e. falsely detecting a target when the target is not present at the location, was often ignored.

In his pioneering work, Koopman [11] employed a negative exponential function of the search effort density to compute the probability of detecting the target. His goal was to allocate the search effort in a way that the probability of detecting the target would be maximized. Later, De Guenin [12] extended Koopman's results to a general detection function applicable to a wider range of applications. Unlike Koopman, he defined the detection probability as a function of the environmental conditions at a location. He illustrated the intuitive effect of environmental conditions on detection probability with a simple example of visual search where the detection probability depends upon the atmospheric conditions that may not be uniform and thus, for the same search effort, the detection probability will be higher in clear areas than in overcast areas.

The search objective in Koopman's work was either to maximize the probability of detecting the target or to minimize the expected time to find it. Note that, here, detecting the target and finding the target implies the same thing since a detection event must necessarily indicate that the target is found. Mela [13] pointed out that maximizing the probability of

detection may not necessarily be the correct search procedure. Rather, the optimal search strategy should depend on the objective of the search mission. He showed a numerical example where the target is known to be contained in one of two locations and an action is to be taken based on the results of two observations acquired from these locations. His results show that the optimal allocation of search effort for this problem is to observe the same location twice, and it is different than the allocation of the search effort based on maximizing the detection probability which assigns single observation to both locations. Mela called this optimal strategy the *correct commitment of forces*. Kadane [14] later formalized Mela's point and called it the *optimal whereabouts search problem*, where he formally defined the goal as maximizing the probability of correctly stating which location contains the target at the end of the search mission.

Another search problem that derives from Koopman's work is Chew's optimal stopping problem [15]. Chew introduced a penalty cost for terminating the search with the case of non-detection and stopped the search when the termination penalty was sufficiently small relative to the cost of additional searching. He pointed that the objective of search in this problem is to minimize the searcher's risk or the total expected cost of searching and stopping. Determining when to optimally stop plays an important role whenever there is not a definite budget on search effort, but the competing costs of possible outcomes are known.

Pollock [16] is the first to consider false positives in a search problem. He considered the scenario where a decision is to be given on the presence or absence of a target at a single location. When a new measurement is available, either the searcher arrives at a decision on the state of the target (target is present or target is absent in the location) based on the acquired information so far, or the search continues. His objective is to minimize the expected cost of the search process. However, his problem did not include the optimal allocation of search effort since he considered a single search location. The first study that is concerned with optimal allocation of search effort in the presence of false positives is conducted by Stone *et al.* [17]. They suggested that a close inspection after a detection event is required to determine with certainty whether the detection event occurs due to the target of interest or a false positive. Thus, in their search strategy, the search effort is continuously split between broad search and contact investigation. When the broad search generates a detection, a

contact investigation starts and the source of the detection is found. If the detection is found to be due to the target, the search stops. Otherwise, the broad search restarts and these two steps, broad search and contact investigation, continue one after the other until the target is found. This strategy is particularly useful if the number of false positives is small and switching between the broad search and contact investigation happens with no additional or negligible cost.

Dobbie [18] extended the results of Stone *et al.* to a more general case where contact investigation can be postponed or even avoided. When the practical requirements of the search problem limit performing immediate contact investigation after each detection event, he suggested book-keeping of all detection events and when and where these detection events occurred. As contact investigation starts and some detection events are identified as false detections, one can use this information to make inference about the remaining detection events based on their locations, densities and time of when they were produced. For example, the acquired information about the location of an identified false detection may be sufficient to accurately anticipate that a nearby detection is also a false detection so that allocating effort to investigate that detection can be avoided. However, in order to increase the accuracy of any inference about the residual detections, it is likely to perform contact investigation for a large number of times, which may degrade the applicability of the procedure to practical search applications.

The existence of false detections is more often considered in recent works compared to the early stages of search theory literature. Kalbaugh [19] addresses the design of search patterns when the target is among Poisson distributed false detections. However, some of his assumptions, e.g. assuming that the process of determining whether a detection event is due to a false positive or it is due to the target of interest can be accomplished without any delay, do not satisfy the practical requirements of many search applications. The first notable paper that addresses the existence of false positives is by Kress *et al.* [20]. Kress *et al.* consider the problem of finding a hostage hidden in one of n possible locations. They assume that a verification team is also present to verify the presence of the hostage at a suspected location and the verification step is not subject to any error. Total search time consists of the time to search for the hostage and the time to rescue the hostage once his/her

location is determined. They show a greedy strategy when the search objective is to minimize the expected time to rescue the hostage. However, their analysis imposes assumptions that can be restrictive in practice. For example, they assume that the time to transit from one location to another is zero, and that search and verification tasks operate concurrently at the same location. In many real-world applications, both the search and the verification tasks are either performed by the same unit or the verification step follows the search. Chung *et al.* [21] also address search with false positives in a very recent paper. The authors propose a formulation of the search problem where a mobile search agent seeks to locate a target in a given region or declare that the target is absent. The authors account for false positives, false negatives, and the transition costs of moving to another search location. The search objective is to minimize the expected time until the decision on a single target's presence or absence is made. However, we note that their approach does not address environmental variability or the presence of multiple targets.

In a vast majority of studies on search theory, the task is to locate a single target. Thus, even in the presence of false detections, the search stops when the target of interest is found or when adequate belief in the absence of the target is obtained. However, in many real-world search applications, it is of particular interest to find multiple targets or to learn the target density in the region. Surprisingly few studies in the search literature address problems that involve multiple targets. Cozzolino [1] is the first to address a multiple targets search problem. He assumed that the number of targets in a search area is a random variable that has a Poisson distribution, and he considered the problem of determining when to optimally stop searching the area. Since only a single search location is considered, his work does not address how to allocate the search effort among possible search locations. Luss [22] also considered a similar problem where the number of targets is unknown but follows either a Poisson distribution or a binomial distribution. Unlike Cozzolino, Luss considered finding the optimal search trajectory that maximizes total returns during a planning horizon subject to effort constraints. Smith and Kimeldorf considered the problem of finding at least one target in many targets [23]. They also assumed a Poisson distribution on the unknown number of targets. In their work, search stops immediately when any one of the targets is found. Smith and Kimeldorf, in a later study, considered the problem of finding all the

targets in the search area in the minimum expected time [24]. However, their assumption that the number of targets follows a Poisson distribution remained the same. They also further assumed that when all targets were found, information about all targets being found would be revealed to the searcher. The derivation of the optimal strategy in these studies strictly depends on the assumed distribution for the number of targets and cannot be easily modified when this assumption is removed. Hence, the proposed strategies in these studies apply to a very specific class of search applications that satisfy the assumption of the known distribution on the number of targets. Recently, Wong *et al.* [25] consider the problem of searching for multiple lost ships in an ocean environment with multiple searchers. They assume that the number of targets (ships) are known with certainty, which significantly simplifies the problem. Lau [26] has offered a search strategy when the number of targets is uniformly unknown and the objective is to minimize the expected time to find all targets. This is the only search problem considering a uniformly unknown number of targets that we are aware of. However, Lau imposes assumptions that limit the practical utility of his work, such as the assumption of perfect measurements (when there is a target in a location, it is found with probability 1). We also note that all of these studies that consider multiple targets neglected the existence of false positives. To the best of our knowledge, there is not any work in the literature where the search problem addresses both the unknown number of targets and the existence of false detections.

The three most commonly employed search objectives in the literature are: maximizing the probability of finding the target [7, 11, 27–30], minimizing the expected time to find the target [17, 18, 20, 21], and minimizing the expected cost of a search procedure [1, 15, 16, 23, 31]. In addition to these search objectives, information-theoretic approaches such as minimizing the entropy of the posterior distribution, or equivalently, maximizing the information content of the posterior distribution are also sometimes employed in search problems [32, 33].

In both search theory and information theory, there is an initial uncertainty about the location of the target. However, while information theory solely aims to reduce this uncertainty, search theory can benefit from additional criteria such as maximizing the probability of correctly guessing the target location. Mela [13] illustrates this point with a few simple examples and he shows that the relation between search theory and information theory is

tenuous. According to Mela, search problems should be regarded as an application of the more general theory of statistical decisions. Barker [33] opposes Mela and proves that in case of an exponential detection function, the search rule that maximizes the probability of detecting the target also minimizes the entropy of the posterior distribution. Thus, he suggests that information theoretic approaches can apply to search problems in the case of exponential detection functions. However, his results are very restrictive and applicable to very few real-world problems. Richardson [32] also suggests the applicability of information-theoretic approaches in search problems and demonstrates that the concepts of information theory are relevant to certain kinds of search and surveillance problems, particularly when false detections are considered. In his notable paper, he compares the results of a policy that is based upon maximizing the information content of the posterior distribution with the results of an optimal single stage look-ahead policy. He shows that maximizing the information gain is the optimal policy when the planning horizon is sufficiently large, and he suggests that information theoretic approaches are suitable in search and surveillance problems that incorporate false detections. However, he tries only a small number of alternative policies for the comparison, which limits the generalization of his results. In addition, we note that such a relation is feasible only when addressing presence or absence of a single target in a location. In general, while the information-theoretic approaches can be applicable to certain types of search problems (for instance, searching for hydrothermal vents [34]) search theory and information theory address fundamentally different problems. In Section 3.6, we show that maximizing the information gain does not necessarily yield the optimal performance.

1.3 Content of this dissertation

In this section, we briefly present the search problems that we consider in this dissertation. We also provide a brief discussion of our approach for each problem.

1.3.1 Single-agent search problem

The search problem we consider involves multiple targets and false positives. An unknown number of targets are randomly distributed in the search area. We consider that the goal of

a search mission is to find the number of targets at each location, and the search mission is subject to a time or distance constraint due to the limited endurance of the search vehicle.

The goal of finding the number of targets at each location is not very different from what Mela [13] and Kadane [14] previously proposed. In an optimal whereabouts search, the goal is to maximize the probability of correctly guessing the target location after the termination of search. Since there is a single target in the search area, for each search location either the location contains the target or the location is empty. Thus, guessing the target location is indeed not very different from guessing the binary number of targets in each location. When multiple targets and false detections are not considered, Kadane's results can be modified without much work to address the problem of finding the number of targets at each location. However, the existence of multiple targets and false detections necessitates the development of new tools to compute optimal search strategies. In this dissertation, we propose decision-theoretic value functions to compute optimal search locations in a search problem that addresses both multiple targets and false positives, and we believe our decision-theoretic value functions successfully apply to many real-world search applications.

We also note that our results agree with Richardson's results [32] when there is at most one target at a location. Richardson compared the performance of a policy based upon maximizing the information content of the posterior distribution with the results of an optimal single stage look-ahead policy, and he showed that an information-theoretic approach yielded the optimal search performance. However, in his problem, if he was to compare the performance of the information-theoretic approach to the performance of our decision-theoretic cost function, he would likely obtain the same result for both. This is because when there is at most one target at a location, the posterior distribution that is more likely to contribute to a correct estimate also has larger information content. On the other hand, when there could be multiple targets at a location, Richardson's results do not apply, and the performance of our approach outperforms the performance of an information-theoretic approach.

Our search problem also addresses the effect of the environment on search results. We consider that the local environmental conditions affect the sensor performance, and the environment varies throughout the search region. Hence, the sensor performs better in some locations than in others. Moreover, the environment at a location may not be known with

certainty, instead we may have a stochastic knowledge of the environment. Another strength of our approach is that it accounts for the transition costs of moving from one location to another, which is largely ignored in the existing literature. Due to the assumed motion constraint on the vehicle, we consider a uniform cost for transition, but our approach can be easily modified to account for non-uniform transition costs, too. To sum up, our work builds upon prior work by accounting for false detections, multiple targets, and uncertainty in the environment. To the best of our knowledge, this is the first study to account for all these factors together.

1.3.2 Characterizing the local environmental conditions

When the environment is uncertain, we may estimate the performance of a search sensor inaccurately. Inaccurate estimates of sensor performance can lead to inaccurate estimates of search performance. For example, when the presumed probability of detection is higher than the actual probability of detection, the probability that all targets have been found during a search mission is exaggerated, and the search mission might be terminated too early. For the particular trial in [35], the experimental results show that the mine-hunting mission takes 40% less time when the environment is known compared to when there is no prior environmental information. When there is an environment sensor that senses the local environmental conditions, the question is where to sense the environment so that the search performance can be improved. In order to compute the optimal locations to acquire environment measurements, we seek to derive a decision-theoretic cost function. How the cost function can be derived largely depends on how the search sensor and the environment characterization sensor operate. We consider several cases where the search sensor and the environmental characterization sensor are either placed on separate vehicles or they are both placed on the same vehicle and derive a decision-theoretic cost function to compute the optimal paths for each case. The considered cases are

1. each sensor is placed on separate vehicles,
2. both sensors are placed on the same vehicle and they operate simultaneously,
3. both sensors are placed on the same vehicle but only one sensor operates at a time.

1.3.3 Multi-agent search problem

We also consider the search scenario where there is more than one search agent, and each agent is equipped with a search sensor. The goal is to collaboratively search the area and maximize the joint search performance. As in any multi-agent search problem, a major challenge is to find a tractable solution that scales well with the number of agents.

One approach that is sometimes used in multi-agent search problems is that the agents are assumed to be capable of broadcasting their history of actions and observations to the other agents at any time. In many search problems this is not a feasible assumption since communication is very limited and agents can communicate with each other only if they are close enough. Another approach is to assume that communication is not possible and each agent plans its path based on its local information [36, 37]. While this approach reduces the computational complexity of the problem, it may significantly degrade the utility of the search and yield suboptimal search performance since different agents may end up searching the same locations. This work considers a decentralized approach where each agent can share its local information with the other agents within a certain communication range, and each agent plans its non-myopic search strategy based on its local information and the information shared with other agents.

Communication is useful when, for example, an agent explores an area with an abundance of targets and informs the other agent so that the other agent abandons its current search path and instead searches the same area. However, in order to communicate with the other agent, the communicating agent must abandon its path and move towards the other agent's communication range, which represents the cost for communication. Thus, the challenge is to determine when an agent should communicate its history of actions and observations with another agent. We refer to this problem as the *when-to-communicate* problem. To achieve the optimal joint search performance, each agent must reason about the actions and observations of other agents, which easily becomes intractable for even a small-size problem. Instead, our approach decouples path planning actions from communication planning actions, which significantly reduces the complexity of the when-to-communicate problem. To tackle the path planning problem, we modestly extend our results from the single-agent search problem to the multi-agent case. Then, given each agent's optimal search path, we use

heuristics to tackle the when-to-communicate problem.

1.4 Dissertation overview

This dissertation offers formal solutions to several search problems where a search sensor and/or an environment characterization sensor are assumed to be available. For each problem, simulation results are provided for a comparison of the proposed approach with the standard methods that are often employed in similar applications. The remainder of this dissertation is organized as follows.

In Chapter 2, we formally define the search problem that we address and derive the decision-theoretic cost function to compute the optimal search locations. Our adaptive mission planning is implemented in a receding horizon strategy in which an optimal finite-length path is continually computed using real-time data as the search agent travels. We compare our approach with a naive mowing-the-lawn approach that is commonly used in many search applications. We show that our adaptive search strategy leads the search agent to explore more informative parts of the search area and maximize the aggregate certainty about the number of targets.

In Chapter 3, we assume the availability of an environment sensor that detects the local environmental conditions at a location. We consider different scenarios where the search sensor and the environment sensor are either placed on separate agents, or both sensors are placed on the same agent and they operate simultaneously. For each scenario, we propose an approach for finding the optimal locations to sense the environment. We compare our approach for each scenario with an entropy-reduction maximization method and present the results of Monte Carlo simulations to show the efficacy of our approach.

In Chapter 4, we consider the scenario where both the search sensor and the environment sensor are placed on the same agent, but only one sensor can be active at a time. We present our approach to optimally determine when and where to search and when and where to characterize the environment so that a desired probability of attaining a desired risk level can be achieved.

In Chapter 5, we extend our results for the single-agent search problem to the multi-

agent case. We present our approach to compute the optimal search paths and optimal communication actions for each agent. We also propose a simple heuristic to reduce the number of computations an agent in a multi-agent problem has to perform.

In Chapter 6, we show the approximate methods for computing near-optimal paths for the problems discussed in Chapter 2 and 3. In Chapter 2 and Chapter 3, we propose approaches to achieve optimal performance. However, scalability of these approaches is a concern for real-time planning. Thus, in Chapter 6, we present methods to compute near-optimal paths in real-time. We test the efficacy of the approximate solvers in a large search environment using real sonar data previously acquired by our collaborators in Boston Harbor.

Finally, in Chapter 7 the contributions of this dissertation are summarized. We also elaborate on possible ways of extending the work reported herein.

Chapter 2

Computing Optimal Search Paths

This chapter introduces the search problem that we address in this thesis. In Section 2.1, we formally define the search problem and state the assumed characteristics of the search sensor. We also show the belief update rule to update the probability distribution of the number of targets after acquiring a search measurement. In Section 2.2, we define two search objectives that we employ to compute the optimal search paths. We briefly discuss when one objective is preferred over the other, and we show the derivation of our decision-theoretic approach for each search objective. The results of the numerical illustrations and a discussion of the effect of environmental uncertainty on search performance are presented in Section 2.4.

2.1 Preliminaries

2.1.1 Problem formulation

We are given a bounded search grid $\mathcal{G} \subset \mathbb{R}^2$ with n_r rows and n_c columns partitioned into $K = n_r n_c$ disjoint cells. We associate with each cell i random variables X_i and E_i that represent the number of targets and the environmental conditions in the cell, respectively. Since environmental conditions predict search sensor performance, we presume that knowledge of environmental conditions is equivalent to knowledge of search sensor performance. We presume X_i is independent of X_j and E_i is independent of E_j when $i \neq j$. The search vehicle's objective is to estimate $X_1, \ldots, X_K$. We use a stochastic description of sensor performance

in the environment.

We assume that the environment in each cell is from a finite set of possible environments $w_1, w_2, \ldots, w_m$. We presume that the actual environmental condition in each cell is not known, but that a probability distribution is known for each cell. The environment probability distribution for the ith cell is expressed $P(E_i) = \big[p_1(i), p_2(i), \ldots, p_m(i)\big]$ where $p_j(i) = P(E_i = w_j)$ is the probability that the environment is w_j. We note that the sum of probabilities for each cell is unity.

$$\sum_{j=1}^{m} p_j(i) = 1 \tag{2.1}$$

2.1.2 Search sensor model

When the search vehicle visits a cell, it acquires a noisy observation $z \in Z$ of the number of targets in the cell. We denote by Z both the set of possible search measurements (i.e. $z \in Z$) and the random variable associated with a search measurement in a cell (i.e. $Z_i = z_i$). The observation z may be less than the true number of targets because of false negatives, or it might be larger due to false positives. We assume that the number of false detections z_f and the number of correct detections z_d are probabilistically independent. Hence, the value of the measurement z can be expressed

$$z = z_f + z_d$$

We model the likelihood of observing z targets when x is the true number of targets given that the environment is w_j. The sensor model is

$$P\big(Z = z \mid X = x, E = w_j\big) = \sum_{l=0}^{\min(x,z)} P_D\big(z_d = l \mid x, w_j\big) P_F\big(z_f = z - l \mid w_j\big) \tag{2.2}$$

where $P_D(z_d = l \mid x, w_j)$ is the probability that the sensor detects l targets, and $P_F(z_f = k \mid w)$ is the probability that the sensor returns k false positives. The convolution of P_D and P_F in (2.2) follows from the assumption that the number of false negatives and the number

of false positives are statistically independent. We model the probability of observing a false positive with a geometric distribution, and the probability of observing a correct detection with a Binomial distribution,

$$P_F\big(z_f = k \mid E = w_j\big) = (1 - F_j)\ F_j^k \qquad k \geq 0 \tag{2.3}$$

$$P_D\big(z_d = l \mid X = x, E = w_j\big) = \binom{x}{l} D_j^l (1 - D_j)^{x-l} \qquad 0 \leq l \leq x \tag{2.4}$$

where $0 \leq F_j < 1$ denotes the probability of one or more false positives, and $0 < D_j \leq 1$ denotes the probability of correct detection. Note that both F_j and D_j are assumed to vary as functions of the environment type w_j. Then, the likelihood is expressed

$$P\big(Z = z \mid X = x, E = w_j\big) = \sum_{k=0}^{\min(x,z)} \binom{x}{k} D_j^k (1 - D_j)^{x-k} (1 - F_j) F_j^{z-k} \tag{2.5}$$

Note that the probability of acquiring true number of targets increases with increasing probability of detection and decreases with increasing probability of false positives. Let x be the true number of targets in a cell. When $D = 1$ and $F = 0$, we acquire perfect measurements of the number of targets.

$$P\big(z = x \mid x, D = 1, F = 0\big) = 1 \tag{2.6}$$

In subsea applications, the probability of acquiring a false positive and the probability of correctly detecting a target are sometimes modeled through a receiver operating characteristics (ROC) curve which describes the probability of correct detection as a function of the probability of false detection [38–40]. We note that our intention in this study is not to model the characteristics of a specific sensor type. We believe the binomial distribution in (2.4) is a natural choice to model the probability of correct detections since the problem involves multiple targets and the number of targets detected at a location is a function of the true number of targets at that location. Similarly, we believe the geometric distribution in (2.3) efficiently models the intuition that fewer false detections are more likely to occur

than a greater number of false detections. However, we note that, other expressions are also possible for modeling the false positives and correct detections, and our approach for computing the optimal search paths do not depend on these specific models. Hence, our results reported in this thesis except for numerical illustrations can be generalized to other expressions for modeling the false positives and correct detections.

2.1.3 Belief update rule for the search vehicle

When the search agent samples from a location, it acquires the noisy measurement of the number of targets. We use the Bayesian update law to update the probability distribution of the number of targets when the search measurement z is observed

$$P\big(X = x \mid Z = z, E = w_j\big) = \frac{P\big(Z = z \mid x, w_j\big)P\big(X = x\big)}{P\big(Z = z \mid w_j\big)} \tag{2.7}$$

where $P(X = x)$ is the prior distribution of the number of targets, and $P(Z = z \mid x, w_j)$ is the sensor characteristics if w_j is the true environment and $P(Z = z \mid x, w_j)$ follows from (2.5). We note that (2.7) is obtained since we assume that the number of targets at a location is probabilistically independent of the environmental conditions at that location, and thus $P(X = x \mid E = w_j) = P(X = x)$. The probability distribution $P(Z = z \mid E = w_j)$ in (2.5) can be computed

$$P\big(Z = z \mid E = w_j\big) = \sum_x P\big(Z = z \mid X = x, E = w_j\big)P\big(X = x\big) \tag{2.8}$$

2.2 Search objectives

We assume that the goal of a search mission is to estimate the number of targets at each search location. Due to the time/distance constraint on the mission, searching every location may not be possible. Instead, we may need to choose a set of locations to sample from. Determining the best sampling locations is the key to achieving optimal search performance, and it is related to how we evaluate the value of a search measurement. In our decision-theoretic framework, the value of a search measurement is associated with how likely it is to

contribute to a good estimate of the number of targets with the acquired measurement. We define two different search objectives to address this goal

1. sample locations that are expected to maximize the probability of correctly estimating the number of targets

2. sample locations that are expected to minimize the penalty due to incorrect estimation of the number of targets

Given the specifics of the search problem, we may be interested in optimizing one of these two search objectives. The first objective is often useful when incorrect estimations are equally bad, and we seek to maximize the probability that our estimate of the number of targets after a mission will be correct. The second objective, on the other hand, is more preferred in search scenarios where we seek to avoid cases of incurring a large penalty for incorrect estimations.

Thus, to address the first objective, we use a zero-one utility function, and to address the second objective, we use a linear-loss function.

2.2.1 Objective 1: maximize estimation accuracy

Suppose we seek to maximize estimation accuracy. After the search vehicle visits a location, we compute the estimate $\delta_X(z)$ of the number of targets x at the location, based on the measurement z. When $\delta_X(z)$ is greater than x, we overestimate the number of targets, i.e. we declare more than the actual number of targets are present. When $\delta_X(z)$ is less than x, we underestimate the number of targets, i.e. we fail to declare some of the targets that are present. Both overestimation and underestimation may degrade the utility of the search results. Given the measured data z, we define the utility of the estimate $\delta_X(z)$ when x is the true number of targets

$$U\big(x, \delta_X(z)\big) = \begin{cases} 1 & \text{if } x = \delta_X(z) \\ 0 & \text{if } x \neq \delta_X(z) \end{cases} \tag{2.9}$$

where all deviations from the true number of targets are penalized equally. We then compute the utility of acquiring the measurement z for environment w_j.

$$\mathbb{E}\Big[U\big(x, \delta_X(z)\big) \mid z, w_j\Big] = \sum_x P\big(X = x \mid z, w_j\big) U\big(x, \delta_X(z)\big) \tag{2.10}$$

The expectation in (2.10) is taken over the parameter space X with respect to the posterior distribution $P(x \mid z, w_j)$. Let $\delta_X^\star(z)$ be the estimator that maximizes the expected utility in (2.10). Such an estimator is called the Bayes estimator, and it is a function of the acquired measurement z

$$\delta_X^\star(z) = \arg\max_{\delta_X(z)} \ \mathbb{E}\Big[U\big(x, \delta_X(z)\big) \mid z, w_j\Big] \tag{2.11}$$

Then, the expected utility in (2.10) is the estimation accuracy conditioned on environment w_j that we seek to maximize, when the estimator $\delta_X(z)$ is the Bayes estimator in (2.11). For the specific utility function in (2.9), the corresponding Bayes estimator is given in (A.3). Thus, the estimation accuracy conditioned on the environment w_j after acquiring the measurement z is

$$\mathbb{E}\Big[U\big(x, \delta_X^\star(z)\big) \mid z, w_j\Big] = \max_x P\big(X = x \mid z, w_j\big) \tag{2.12}$$

In order to assess the benefit of searching a cell, we compute estimation accuracy in (2.12) for each possible measurement $z \in Z$. This yields *expected* estimation accuracy of searching a cell conditioned on environment w_j,

$$\mathbb{E}\Big[U\big(x, \delta_X^\star(z)\big) \mid w_j\Big] = \sum_z P\big(Z = z \mid w_j\big) \max_x P\big(X = x \mid z, w_j\big) \tag{2.13}$$

where $P\big(Z = z \mid w_j\big)$, the probability of observing a particular measurement given the true environment w_j, is defined in (2.8). Since deterministic knowledge on the environment is assumed to be unavailable, we compute the expected estimation accuracy unconditional on the environment. Averaging over the environments yields

$$\mathbb{E}\Big[U\big(x,\delta_X^\star(z)\big)\Big] = \sum_{j=1}^{m} P\big(E=w_j\big)\mathbb{E}\Big[U\big(x,\delta_X^\star(z)\big)\mid w_j\Big] \tag{2.14}$$

and we call this the *anticipated* estimation accuracy.

Measurements from different cells are independent, and thus estimation accuracy for a path that passes through multiple cells is simply a product of the estimation accuracy for each cell in the path. Let $\gamma = [q_1, q_2, \ldots, q_N]$ be a candidate search path that traverses the cells $q_1, \ldots, q_N \in \mathcal{G}$, and let $\bar{\mathcal{C}}_\gamma$ be the budget constraint on the search mission due to limited time/distance the vehicle can traverse. We define $\mathcal{C}(\gamma)$ to denote the cost for traversing a path γ. Note that when the traversal cost for moving from one location to another is unity, $\mathcal{C}(\gamma)$ is simply the number of cells traversed by γ.

When the vehicle makes multiple visits to a cell, we acquire a set of independent search measurements. We denote by z both a single search measurement and a set of search measurements when a cell is visited multiple times by γ. Then, the expected utility of traversing γ is

$$\mathbb{E}\Big[U\big(x,\delta_X^\star(z_\gamma)\big)\Big] = \prod_{q_i\in\gamma}\mathbb{E}\Big[U\big(x_{q_i},\delta_X^\star(z_{q_i})\big)\Big]\prod_{i\in\mathcal{G}\setminus\gamma}\max_{x_i} P\big(X_i=x_i\big) \tag{2.15}$$

where $\mathcal{G}\setminus\gamma$ denotes the remaining cells in the search grid that are not traversed by γ, and $\max_{x_i} P\big(X_i=x_i\big)$ is the certainty in the number of targets in cell i prior to acquiring new measurements. Let Ω_γ denote the finite collection of feasible search paths. Then, the optimal search path is

$$\gamma^\star = \arg\max_{\gamma\in\Omega_\gamma}\mathbb{E}\Big[U\big(x,\delta_X^\star(z_\gamma)\big)\Big] \tag{2.16}$$

subject to

$$\mathcal{C}(\gamma) \le \bar{\mathcal{C}}_\gamma \tag{2.17}$$

2.2.2 Objective 2: minimize the penalty due to incorrect estimations

Now, suppose we seek to minimize a penalty associated with incorrectly estimating the number of targets. We define that the penalty increases linearly with the distance between the true number of targets and its estimate. Thus, given the measured data z, we define the linear loss corresponding to the estimate $\delta_X(z)$ when x is the true number of targets

$$L_X\big(x, \delta_X(z)\big) = \begin{cases} c_1\big(x - \delta_X(z)\big) & \delta_X(z) < x \\ c_2\big(\delta_X(z) - x\big) & \delta_X(z) \geq x \end{cases} \tag{2.18}$$

where $c_1 > 0$ and $c_2 > 0$ are relative costs of overestimating $(\delta_X(z) > x)$ and underestimating $(\delta_X(z) < x)$ the number of targets. For some applications, such as mine-hunting and search and rescue, overestimating the number of targets is preferred to underestimation. In mine-hunting missions, overestimating the number of mines, due to false positives, may lead to wasted follow-on effort in neutralizing mines that are not present. However, underestimating the number of mines may have disastrous consequences. Thus, we may assign the relative costs such that $c_1 > c_2$.

Similar to our derivation in Section 2.2.1, we compute the posterior expected loss of forming the estimate $\delta_X(z)$ after acquiring the search measurement z when the environment is w_j

$$\mathbb{E}\Big[L_X\big(x, \delta_X(z)\big) \mid z, w_j\Big] = \sum_x P\big(X = x \mid z, w_j\big) L_X\big(x, \delta_X(z)\big) \tag{2.19}$$

and the Bayes estimator is

$$\delta_X^\star = \arg\min_{\delta_X(z)} \ \mathbb{E}\Big[L_X\big(x, \delta_X(z)\big) \mid w_j\Big] \tag{2.20}$$

Expected loss in (2.19) is called Bayes' risk when $\delta_X(z)$ is the Bayes estimator in (2.20). For the specific loss function in (2.18), the corresponding Bayes estimator is given in (A.5). For notational convenience and clarity of presentation, we define two types of risk associated with the value of searching a location: the *conditional current risk* and the *conditional antic-*

ipated risk. Loosely speaking, the conditional current risk is the risk of incorrectly estimating the number of targets with the information at hand, and the conditional anticipated risk is the risk we expect to attain with the additional information after searching the location, both conditioned on the environment w_j. The conditional current risk for the ith cell is denoted

$$\rho(i \mid w_j) = \mathbb{E}\Big[L_X\big(x, \delta_X^\star\big) \mid w_j\Big] \tag{2.21}$$

and the conditional anticipated risk for visiting the ith cell l times is denoted by

$$r(i, l \mid w_j) = \sum_{z_{i,1}} \cdots \sum_{z_{i,l}} P\big(z_i \mid w_j\big)\mathbb{E}\Big[L_X\big(x, \delta_X^\star(z_i)\big) \mid z_i, w_j\Big] \tag{2.22}$$

where $z_i = [z_{i,1}, z_{i,2}, \ldots, z_{i,l}]$ is the set of independent search measurements acquired at cell i. Note that both the conditional current risk in (2.21) and the conditional anticipated risk in (2.22) are conditional on the environment w_j.

The value of acquiring a search measurement at a location is the reduction in uncertainty associated with not knowing the true number of targets at that location due to the acquired measurement. Thus, the benefit of searching a location for l times given the environment at the location is the difference between the conditional current risk and the conditional anticipated risk, and we call this the *attained risk reduction*.

$$B\big(i, l \mid w_j\big) = \rho(i \mid w_j) - r(i, l \mid w_j) \tag{2.23}$$

and, when $l = 1$, the attained risk reduction for a single search visit is denoted $B\big(i \mid w_j\big)$.

The benefit of traversing the candidate search path $\gamma = [q_1, q_2, \ldots, q_N]$ is the total attained risk reduction along the path

$$B\big(\gamma\big) = \sum_{q_i \in \gamma} \sum_{j=1}^{m} P\big(E_{q_i} = w_j\big) B\big(q_i, l_{q_i} \mid w_j\big) \tag{2.24}$$

where l_{q_i} is the cell multiplicity of q_i in path γ. Then, the optimal path is the one that maximizes (2.24) subject to the budget constraint in (2.17)

$$\gamma^{\star} = \arg\max_{\gamma \in \Omega_{\gamma}} B\big(\gamma\big) \tag{2.25}$$

2.3 Notes on the search environment and the search vehicle

The search objectives from Section 2.2.1 and Section 2.2.2 offer a rigorous way of computing the optimal search paths for a general search mission. When a search mission is performed in a particular search environment, the mission is constrained by the properties of that environment as well as the properties of the search vehicle that performs the mission. Throughout this dissertation, our numerical illustrations are inspired by subsea search applications, such as mine-hunting missions. Thus, when we evaluate the performance of a proposed approach, we consider a search mission that satisfies the typical properties and requirements of a subsea search application.

2.3.1 Search environment

We assume that a bounded search area $\mathcal{G}$ is divided into a grid with K non-overlapping cells. For each cell i, we assume there is $0 \leq x(i) \leq L$ number of targets bounded by an upper bound L. In mine hunting missions, $x(i)$ represents the number of mines residing in cell i. We assume no prior information exists about the number of targets in any cell. That is, we consider a uniform probability distribution of the number of targets where each possibility $x = 0, 1, \ldots, L$ has equal prior probability. We note that L is typically not known beforehand; however, letting L be a sufficiently large number will capture every possible scenario. We assume there is a finite number of possible environments $w_1, \ldots, w_m$ in the search area. In all our simulations in this dissertation, we consider that $L = 2$ and $m = 3$. That is, we consider that possible number of targets in a cell is $x = 0$, $x = 1$, or $x = 2$, and possible environments in the search area are w_1, w_2, and w_3.

The performance of the search sensor is dependent on the environmental conditions. The particular sensor model that we use throughout this dissertation is (2.5). However, it

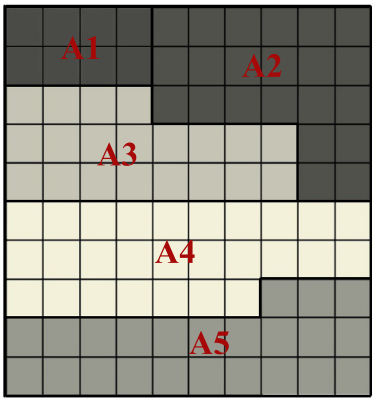
should be noted that our decision-theoretic approaches do not depend on a particular sensor model and any other sensor model can also be used to assess the search performance. We assume that for each cell in the search area, a corresponding probability distribution of the environments $\Pi = [p_1, p_2, p_3]$ is given, where p_j is the probability that the environment is w_j. For example, when $\Pi = [0.15, 0.2, 0.65]$ for a cell, there is 0.15 probability that the environment is w_1, 0.2 probability that the environment is w_2, and 0.65 probability that the environment is w_3 in that cell.

2.3.2 Search vehicle

We consider that the search vehicle is equipped with a side-scan sonar sensor that is often used in subsea applications. The side-scan sonar sensor images the seafloor and detects the targets in a location, that yields a noisy search measurement of the number of targets at that location. Due to the characteristics of a side-scan sonar sensor – the perturbations in the straight-line motion of the vehicle significantly distort the quality of the data acquired from a side-scan sonar sensor (see, for example, [41, 42]) – we assume that the sensor operates only when the vehicle is moving in a straight line. Therefore, we usually assume that the vehicle can only move forward to the next grid cell in a row and it can transit between different rows only outside of the search domain. The resulting path consists of parallel straight lines. We associate a unit cost for moving from a cell to an adjacent cell. Thus, the budget constraint on the search mission is simply the number of cells the vehicle can visit during a search mission. We refer to this number as the *mission length*, and we denote it by $\mathcal{N}$. The vehicle turns are associated with the same cost as moving forward to the next cell, but passing cells that are outside the search area do not improve search performance since no measurements are acquired, which implicitly penalizes the traversal of cells that are outside of the search area. Given a mission length $\mathcal{N}$ for the search mission, the $\mathcal{N}$-length search trajectory contains grid cells and possible transition cells.

2.4 Numerical results

In this section, we present simulation results that show the efficacy of adaptively selecting search paths, given stochastic assumptions on the environment and sensor performance. Without loss of generality, we consider that the objective of the search mission is to maximize the probability that our estimate of the number of targets is correct. Hence, we seek to find the search path that maximizes (2.16). We compare the performance of our adaptive path planning approach to the performance of a mowing-the-lawn approach where the vehicle travels back and forth until the mission length is met. The lawnmower pattern often arises in subsea applications, and in this study it is employed as a baseline against which to assess the efficacy of the adaptive path planning.



Region	Probability distribution		
	w_1	w_2	w_3
A1	[0.90	0.05	0.05]
A2	[0.85	0.10	0.05]
A3	[0.10	0.10	0.80]
A4	[0.00	0.10	0.90]
A5	[0.20	0.70	0.10]

Figure 2.1: Search area and cell-wise environment distributions

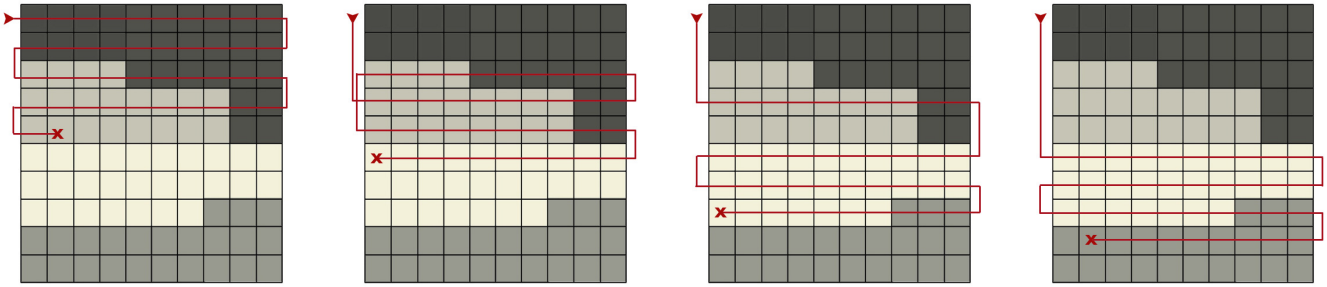
The assumptions on the search environment and the search vehicle are given in Section 2.3. Fig. 2.1 shows a search area that is partitioned into regions A1 through A5. For each region, the corresponding probability distribution on the environments w_1, w_2 and w_3 is also given. Throughout the dissertation, we consider that the probability of detection and the probability of at least one false alarm for each environment are $D = 0.65$ and $F = 0.4$ for environment w_1, $D = 0.8$ and $F = 0.3$ for environment w_2, and $D = 0.95$ and $F = 0.05$ for environment w_3 (also shown in Table 2.1). Note that the information about the number of objects revealed after searching a cell increases with increasing probability of detection

and decreases with increasing probability of false alarm. Thus, environment w_1 is the least and environment w_3 is the most informative. That is, for example, when sampling from a cell with environment w_3, we expect to learn more about the number of targets in that cell compared to sampling from a cell with environment w_1 or sampling from a cell with environment w_2. We consider that the mission length for the search mission is 50.

Table 2.1: Environment types

Environment	Probability of detection (D)	Probability of false alarm (F)
w_1	0.65	0.4
w_2	0.8	0.3
w_3	0.95	0.05

We conduct eight different numerical tests. We first employ a naive mowing-the-lawn approach as a baseline against which to assess the efficacy of the adaptive path planning strategy. Fig.2.2a shows the search trajectory when mowing-the-lawn approach is employed. The red line represents the search vehicle's path. The red cross is the end cell of the path. The search vehicle travels back and forth until the mission length is met. For the other tests, we employ the adaptive search strategy for varying path lengths. Fig.2.2b through Fig.2.2d show optimal $\mathcal{N}$-length receding horizon search trajectories when the path length $\mathcal{N}$ is 7 (Fig.2.2b), when $\mathcal{N}$ is 15 (Fig.2.2c), and when N is 25 (Fig.2.2d). Corresponding expected estimation accuracy for each test is given in Table 2.2. For convenience, we compute the negative logarithm of expected estimation accuracy (2.13) and sum over each cell. Note that the negative logarithm of expected estimation accuracy increases as expected utility decreases, i.e. smaller negative log of expected estimation accuracy implies better search performance. The results show that the adaptive search strategy yields better expected estimation accuracy when compared to the mowing-the-lawn approach. It is also seen that search performance doesn't necessarily improve when path length increases. For example, expected estimation accuracy when path length is 15 is better than expected estimation accuracy when path length is 25. It can be computationally too expensive to compute the optimal paths when the planning horizon is large, and in Chapter 6 we present efficient algorithms to compute near-optimal paths in feasible time.



(a) (b) (c) (d)

Figure 2.2: Results for mowing-the-lawn search and adaptive search for different path lengths. (a) planned path with mowing-the-lawn search (b) and adaptive search when path length is 7 (c) path length is $15, 50$ (d) path length is 25.

Table 2.2: Search performance

Path length	Expected estimation accuracy (negative log)[1]
Mowing-the-lawn (MTL)	82.2
Receding horizon with $N = 7$	77.4
Receding horizon with $N = 15$	74.9
Receding horizon with $N = 25$	76.7
Entire planning horizon ($N = 50$)	74.9

We conduct Monte Carlo simulations to observe variation in estimation accuracy for each search trajectory given in Fig. 2.2. We iterated for 10,000 times to eliminate the effects due to the random nature of observations and plotted the results in Fig. 2.3. The results show that the mean of the random observations for each search trajectory matches the expected estimation accuracies given in Table 2.2. Note that when the mean for a search trajectory increases, the variation in estimation accuracy also increases. This is because when the vehicle samples from less informative parts of the search area the observations have more variation, which yields more variation in estimation accuracy.

[1]negative logarithm of expected estimation accuracy increases as expected utility decreases, i.e. a smaller negative log of expected estimation accuracy implies better search performance

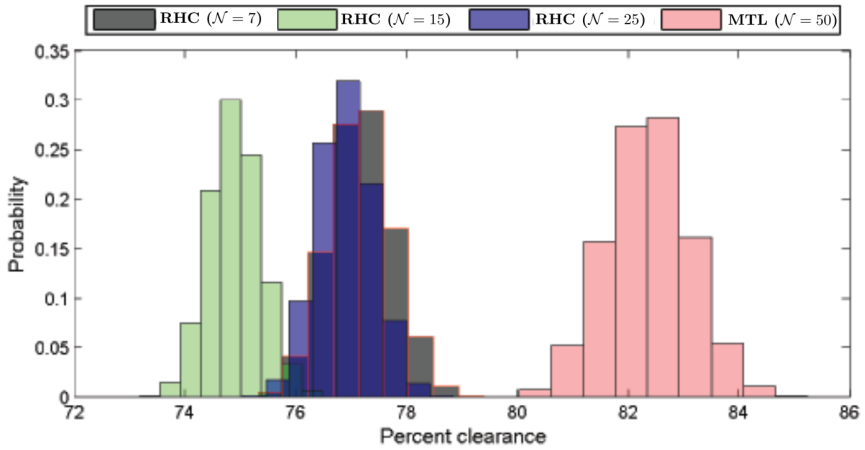


Figure 2.3: Frequency of estimation accuracy for mowing-the-lawn (MTL) and adaptive search with receding horizon strategy (RHC) for different path lengths

2.4.1 Effect of environmental uncertainty on search performance

We now analyze the effect of environment uncertainty on search performance. The observation model for the search sensor is dependent on the local environment. Therefore, uncertainty in the observation model should lead to uncertainty in the posterior distribution of the number of targets which causes the search performance when the environment is uncertain be different than the search performance when the environment is known. We seek a method of formally characterizing the effect of environment uncertainty on the deviation in expected estimation accuracy. For each cell $i \in \mathcal{G}$, let $[p_1(i), p_2(i), \ldots, p_m(i)]$ be the given environment distribution and e_i be the true environment. For notational convenience, let $V(\cdot)$ denote expected estimation accuracy under uniform prior $P(X_i = x_i)$, e.g. $V(e_i) = \mathbb{E}\big[\max_{x_i} P(X_i = x_i \mid z_i, e_i)\big]$. We define μ to be the total deviation from the expected estimation accuracy when the true environment for each cell is known.

$$\mu = \sum_{i=1}^{K} \Big| \, V\Big(e_i\Big) - V\Big(\Pi(i)\Big) \, \Big| \tag{2.26}$$

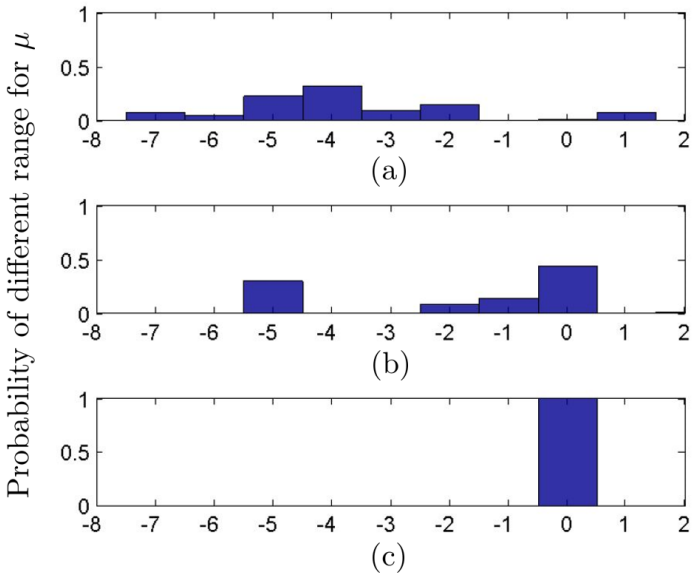


Figure 2.4: Frequency of deviation from actual expected estimation accuracy for different ranges of μ. (a) when $16 \leq \mu \leq 18$, (b) when $10 \leq \mu \leq 12$, (c) when $3 \leq \mu \leq 5$

As the certainty in the environment increases, μ in (2.26) is reduced and consequently we expect that planned expected estimation accuracy before a mission better matches actual estimation accuracy after a mission. We say we underestimate the environment if $V(e) > V(\Pi)$, and overestimate if $V(e) < V(\Pi)$. While underestimating the environmental conditions may decrease search efficacy, overestimation may yield inaccurate decisions on number of targets. In subsea applications, the consequences of the latter case might be severe. In [35], the authors conduct mowing-the-lawn experiments first when there is uncertainty in the environment, and later when the environmental conditions are deterministically known. They show that deterministic knowledge of the environment shortens the mission duration and improves search performance.

In order to numerically evaluate the effect of environment uncertainty on search performance, we conduct Monte Carlo simulations where the environment in each cell is randomly drawn from the environment distributions in Fig. 4.1. We analyse how expected estimation accuracy after a search mission changes when μ in (2.26) decreases. We define *actual* expected estimation accuracy to be expected estimation accuracy that we in fact achieve

instead of expected estimation accuracy assumed to be achieved when environment is uncertain. Fig. 2.4 shows that as μ decreases, the difference between actual expected estimation accuracy and expected estimation accuracy when environment is uncertain decreases. This shows that when μ is closer to zero, our search results provide more reliable information of the number of targets.

Chapter 3

Search and Environment Characterization

In this chapter, we assume the availability of an environment sensor that partially detects the local environmental conditions at a location. We consider several scenarios where the search sensor and the environment sensor are either placed on separate vehicles, or they are placed on the same vehicle. When the sensors are placed on the same vehicle, we specifically consider that both sensors operate simultaneously, and thus, sampling from a location returns with both the search measurement and the environment measurement. The discussion for the case where the sensors are placed on the same vehicle but only one sensor can be active at a time is deferred to Chapter 4. In Section 3.1, we provide a brief review of the literature on similar problems. In Section 3.2, we show the environment sensor model and the update rule to update the probability distribution of the environments after acquiring an environment measurement. In Section 3.3, we present our approach for determining the optimal locations to sense the environment when the search sensor and the environment sensor are placed on separate vehicles. In Section 3.4 we present our approach when the search sensor and the environment sensor are placed on the same vehicle *and* they operate simultaneously. In Section 3.5, we show an approximation approach to compute near-optimal paths in feasible time. We show the results of the numerical illustrations to evaluate the efficacy of the proposed approaches in Section 3.6.

3.1 Related work

We address the case that stochastic knowledge of the environment can be acquired, and we describe where the environment should be surveyed in order to improve overall search performance. One approach for selecting where to acquire environmental information is simply to characterize the locations that are expected to yield the greatest reduction of uncertainty about the environment. In other words, one might seek to maximize change in the entropy, which is often employed in similar applications [43–45]. In contrast, a primary contribution of this study is to show that environmental information should be acquired at the locations where the greatest reduction of uncertainty in search performance is expected to occur.

The effect of the environment on search performance is well-known. In subsea applications where sonar is used for search, variations in the seabed induce significant variation in the observed number of false positives and the number of false negatives [46, 47]. For terrestrial applications using ground penetrating radar, search effectiveness is dependent on background clutter and soil properties [48–50]. A few studies in the literature aim to evaluate the benefit of reducing the uncertainty in the environment [35]. In Section 2.4.1, we show that inaccurate estimates of sensor performance can lead to inaccurate estimates of search performance which degrades the utility of the search.

We note that the existing literature on robotic exploration (see [51–54], among many examples) provides little insight to the applications we address. Robotic exploration addresses the challenge of building a map. In contrast, we seek to characterize a subset of the environment with respect to search sensor performance for the goal of improving search effectiveness.

3.2 Preliminaries

Search and environmental characterization are accomplished using different sensors that can be mounted on different vehicles or on the same vehicle. When the sensors are placed on different vehicles, the vehicle that possesses the search sensor is called *the search vehicle* and the vehicle that possesses the environmental characterization sensor is called *the environmen-*

tal characterization vehicle. When the search sensor and the environment characterization sensor operate simultaneously on a single vehicle, we informally refer to the vehicle as *the search/environmental characterization vehicle.*

3.2.1 Environment sensor model

We consider that the local environmental conditions affect the performance of the search sensor, and we are given the finite set of possible environments $w_1, w_2, \ldots, w_m$ that may occur in the search area. Each environment w_j assigns a particular set of the probability of detection, D_j, and the probability of false positives, F_j. That is, each environment represents different operating points for the search sensor. When the environment at a location is sampled, a noisy observation of the true environment at that location is acquired. We assume the likelihood of observing a particular environment conditioned on the true environment is known before the mission starts and it does not change. Insight on the form of the likelihood function arises from research on subsea bottom-type characterization, such as in [55].

3.2.2 Belief update rule for the environment vehicle

When the characterization vehicle characterizes the environment at a location, it acquires the noisy observation $y \in Y$ of the true environment in the cell. We denote by Y both the set of possible environment measurements (i.e. $y \in Y$) and the random variable associated with an environment measurement at a cell (i.e. $Y_i = y_i$). We use a Bayesian update law to update the probability distribution of the environments when y is observed,

$$P\big(E = w_j \mid y \in Y\big) = \frac{P\big(Y = y \mid E = w_j\big) P\big(E = w_j\big)}{P\big(Y = y\big)} \tag{3.1}$$

where $P(E = w_j)$ is the prior probability that the environment at the location is w_j, and

$$P\big(Y = y\big) = \sum_{j=1}^{m} P\big(Y = y \mid E = w_j\big) P\big(E = w_j\big) \tag{3.2}$$

3.2.3 Belief update rule for the search/environmental characterization vehicle

When the search sensor and the environmental characterization sensor operate simultaneously on a single vehicle, the noisy observations $z \in Z$ and $y \in Y$ are acquired simultaneously due to simultaneously activating both sensors. We update the probability distribution of the number of targets with the observed measurement z as in (2.7) and the probability distribution of the environments with the observed measurement y as in (3.1). Then, the updated probability distribution of the number of objects unconditioned on the environment is

$$P\big(X = x \mid z \in Z, y \in Y\big) = \sum_{j}^{m} P\big(X = x \mid z, w_j\big) P\big(E = w_j \mid y \in Y\big) \tag{3.3}$$

where the posterior distributions $P(X = x \mid z, w_j)$ and $P(E = w_j \mid y \in Y)$ follow from (2.7) and (3.1), respectively.

3.3 Sensors operate on separate vehicles

The primary objective of environmental characterization is to improve search performance. With additional information about the environment at a few locations, it might be possible to avoid searching locations where the sensor performs poorly in favor of places where the sensor performs well. We consider two specific cases: 1) environmental characterization is performed prior to search, 2) environmental characterization and search are performed simultaneously. In all cases, we assume that both environmental characterization and search cannot be performed exhaustively due to limited resources. We derive the cost function for case 1 and then show that case 2 is a slight modification of case 1.

Suppose environmental characterization precedes search. It is intuitively appealing that locations for environmental characterization are selected to directly improve search performance. That is, for example when the search objective is to maximize the estimation accuracy in (2.16), we assess the benefit of obtaining a particular environment measurement y at a location by computing the conditional expected utility $\mathbb{E}\Big[U\big(x, \delta_X(z)\big) \mid y \in Y\Big]$. To

assess the expected benefit of a future environmental measurement, we average over all possible environment measurements. However, we see directly that the result is the iterated expectation, and that the effect of environmental samples has been averaged out

$$\mathbb{E}_y\Big[\mathbb{E}_z\big[U\big(x,\delta_X(z)\big)\mid y\in Y\big]\Big]=\mathbb{E}_z\Big[U\big(x,\delta_X(z)\big)\Big] \tag{3.4}$$

Thus, plans for environmental characterization do not directly improve the expected performance of a search plan. We note that this also applies to the case when the search objective is to minimize the risk due to incorrect estimation of the number of targets (2.25). Next, we will briefly introduce the entropy change maximization method that is often used in similar applications, and show why the environment characterization problem considered in this study cannot be addressed with the entropy method. Then, we show our decision-theoretic approach to determine the environment characterization locations.

3.3.1 Entropy change maximization

When environment information can be acquired only in some locations due to limited resources, the question is to determine where to optimally sample the environment. One approach that is often used in similar applications is to maximize the change in entropy due to acquired environment measurements [43–45]. We briefly describe a typical entropy approach for selecting where to sample the environment so that we can compare it to our proposed approach.

Let $H(E=w_j)$ denote the prior entropy of the probability distribution $P(E=w_j)$ and let $H(E=w_j\mid y\in Y)$ be the posterior entropy after acquiring the environment measurement y

$$H\big(E=w_j\big)=-\sum_{j=1}^{m}P\big(E=w_j\big)\log P\big(E=w_j\big) \tag{3.5}$$

$$H\big(E=w_j\mid y\in Y\big)=-\sum_{j=1}^{m}P\big(E=w_j\mid y\in Y\big)\log P\big(E=w_j\mid y\in Y\big) \tag{3.6}$$

Then, the expected amount of change in the entropy for a future environment measurement y can be computed by

$$J\big(E\big) = H\big(E = w_j\big) - \sum_{y} H\big(E = w_j \mid y \in Y\big) \tag{3.7}$$

Let η be a candidate path for the environment characterization vehicle, Ω_η be the finite collection of feasible paths, and $\bar{\mathcal{C}}_\eta$ be the budget constraint on the environment characterization mission. Then, the best path to characterize the environment based on the entropy change maximization method is

$$\eta^\star = \arg\max_{\eta \in \Omega_\eta} J_\eta(E) \tag{3.8}$$

subject to

$$\mathcal{C}(\eta) \leq \bar{\mathcal{C}}_\eta$$

However, we note that the purpose of environmental characterization in this study is not to explore the environment, but to improve the performance of a follow-on search mission. We show in Section 3.6 via numerical studies that maximizing change in entropy does not maximize the performance of follow-on search missions. In order to assess the value of acquiring an environmental measurement such that the performance of a follow-on search mission will be improved, a fundamentally different approach is needed. Next, we present our decision-theoretic approach that directly accounts for the expected improvement in search performance.

3.3.2 Environmental loss function

Due to the uncertainty in the environment and the noise in environmental observations, estimation accuracy after visiting a cell in (2.12) may be different than actual estimation accuracy if the true environment were unambiguously known. In Chapter 2, we discuss

the effect of environment uncertainty on search results, and show that deviations from the true environment result in deviations from actual estimation accuracy and degrade search performance. In this chapter, we extend our findings in Chapter 2 to select the best locations to conduct environmental surveys. Our approach is to define a linear loss function that penalizes deviations from the actual expected estimation accuracy for each cell.

To formally define the loss function, we first introduce a preference ordering $\preceq$ on environments. Suppose there is a finite set of environments $w_1, w_2, \ldots, w_m$. For notational convenience, let $V\big(w_j\big)$ denote expected search performance conditioned on environment w_j. That is, when the search objective is to maximize estimation accuracy, $V\big(w_j\big)$ follows from (2.13)

$$V\big(w_j\big) = \mathbb{E}\Big[U\big(x, \delta^\star(z)\big) \mid E = w_j\Big] \tag{3.9}$$

and, when the search objective is to minimize the risk due to incorrect estimation, $V\big(w_j\big)$ for cell i follows from (2.23)

$$V\big(w_j\big) = B\big(i \mid w_j\big) \tag{3.10}$$

We say the environment w_k is more preferred for the search than the environment w_j if the expected estimation accuracy conditioned on w_k is greater than the expected estimation accuracy conditioned on w_j. That is, we say that $w_k \preceq w_j$ if and only if $V(w_k) \leq V(w_j)$. If for some w_k, w_j when $k \neq j$ we have $V(w_k) = V(w_j)$, then $w_k = w_j$. If $V(w_k) \neq V(w_j)$, we say w_k and w_j are distinct environments. Suppose the environments $w_1, \ldots, w_m$ are distinct and *ordered* so that $w_1 \prec w_2 \prec \ldots \prec w_m$, let e be the true environment in a cell, and let $\delta_E(y)$ be an estimate of the environment based on measurement y and the prior distribution of the number of targets $P\big(X = x\big)$. When the true environment is e, the loss due to the estimate $\delta_E(y)$ is defined

$$L_V\big(e, \delta_E(y)\big) = \begin{cases} c_1\Big(V\big(e\big) - V\big(\delta_E(y)\big)\Big) & \text{if } \delta_E(y) \preceq e \\ \\ c_2\Big(V\big(\delta_E(y)\big) - V\big(e\big)\Big) & \text{if } \delta_E(y) \succ e \end{cases} \tag{3.11}$$

where $c_1, c_2 > 0$ are the relative costs of over and underestimation. Underestimating the environment, $\delta_E(y) \prec e$, may result in unnecessary extra visits to improve the belief of the number of targets at a location. However, overestimating the environment, $\delta_E(y) \succ e$, may yield to inaccurate estimates of the number of targets. In some search applications, such as mine-hunting, overestimation is less preferred to underestimation. Thus, we may assign the relative costs such that $c_1 < c_2$.

Given the environment measurement y, the posterior expected loss of computing the environment estimate $\delta_E(y)$ is

$$\mathbb{E}\Big[L_V\big(e, \delta_E(y)\big) \mid Y = y\Big] = \sum_{j=1}^{m} P\big(E = w_j \mid y \in Y\big) L_V\big(w_j, \delta_E(y)\big) \tag{3.12}$$

where $P\big(E = w_j \mid y \in Y\big)$ is the updated probability that w_j is the environment at the location after observing the environmental measurement y. We choose the estimator that minimizes the expected loss in (3.12). Let $\delta_E^\star(y)$ be the Bayes estimator such that

$$\delta_E^\star(y) = \arg\min_{\delta_E(y)} \mathbb{E}\Big[L_V\big(e, \delta_E(y)\big) \mid y\Big] \tag{3.13}$$

For the specific loss function in (3.11), the Bayes estimator that minimizes the expected loss is given in (A.5).

3.3.3 Path planning for the case characterization precedes search

A benefit of environmental surveys is to reduce the error in anticipated search performance due to uncertainty in the environment. When a location is not visited by the search vehicle during a search mission, acquiring an environment measurement at that location will not affect search performance. From the loss function in (3.11), computing the estimate (3.13) of the environment after acquiring environment measurement y yields the conditional expected loss

$$\mathbb{E}\Big[L_V\big(e,\delta_E^\star(y)\big)\mid y\in Y\Big]=\sum_{j=1}^{m}P\big(E=w\mid y\in Y\big)\;L_V\big(w_j,\delta_E^\star(y)\big) \tag{3.14}$$

that quantifies the amount of uncertainty in anticipated estimation accuracy after acquiring y. Informally speaking, the prior loss before acquiring an environment measurement represents the prior uncertainty, and the conditional expected loss in (3.14) represents the posterior uncertainty in search performance. For notational convenience, we define $\mathcal{R}(y)$ to denote the reduction of uncertainty in anticipated estimation accuracy due to environment measurement y

$$\mathcal{R}(y)=\mathbb{E}\Big[L_V\big(e,\delta_E^\star\big)\Big]-\mathbb{E}\Big[L_V\big(e,\delta_E^\star(y)\big)\mid y\in Y\Big]$$

Then, the gain of acquiring an environment measurement y_i in cell i is the reduction of uncertainty in anticipated estimation accuracy given that cell i is visited by the search vehicle

$$G\big(y_i\big)=\mathbf{I}\big(i,\gamma^\star(y_i)\big)\;\mathcal{R}(y_i) \tag{3.15}$$

where the notation $\gamma^\star(y)$ denote the best path for the search vehicle when the probability distribution of the environment is updated with the acquired measurement y, and the indicator function $\mathbf{I}\big(i,\gamma(y_i)\big):y_i\to[c',1]$ is defined

$$\mathbf{I}\big(i,\gamma(y_i)\big)=\begin{cases}1 & i\in\gamma(y_i)\\ c' & i\notin\gamma(y_i)\end{cases} \tag{3.16}$$

where $0\le c'<1$ is a parameter to determine the relative gain of sampling the environment at locations that will not be searched. Without loss of generality, we consider that $c'=0$.

Let $\eta=\{q_1,q_2,\ldots,q_M\}$ be a candidate path for the environment characterization vehicle and recall that Ω_η is the finite collection of feasible characterization paths. We denote by y both a single environment measurement and a set of independent environment measurements when a cell is visited multiple times by η. Then, the expected characterization gain of traversing η is

$$\mathbb{E}\Big[G_\eta\Big] = \sum_{y_\eta} P\big(Y_\eta = y_\eta\big) \sum_{q_i \in \eta} \mathbf{I}\big(i, \gamma^\star(y_i)\big)\ \mathcal{R}(y_{q_i}) \tag{3.17}$$

and the optimal path is

$$\eta^\star = \arg\max_{\eta \in \Omega_\eta} \mathbb{E}\Big[G_\eta\Big] \tag{3.18}$$

subject to

$$\mathcal{C}(\eta) \leq \bar{\mathcal{C}}_\eta \tag{3.19}$$

Computing the optimal path for the environment characterization vehicle according to (3.18) can be computationally very expensive. In Section 3.5 , we present an approach to approximate the solution of (3.18). The approximate solution significantly reduces the problem complexity.

3.3.4 Path planning when both vehicles operate simultaneously

Path planing for environmental characterization for the case that environmental characterization and search are accomplished by different vehicles that operate at the same time is addressed similarly to the case in Section 3.3.3 where environmental characterization precedes search. We again compute the gain of acquiring environment measurement y_i in cell i as in (3.15). However, the indicator function in (3.16) is modified to account for the possible situations where the search vehicle visits a location prior to the environmental characterization vehicle, in which case environmental characterization cannot influence search plans. The indicator function in (3.16) is unity when the location to be characterized is in the search vehicle's trajectory even if the environmental characterization vehicle arrives *after* the search vehicle. To overcome this problem, we introduce an indexing for each location. Let cell i appear in both vehicles' paths and let $n_\eta(i)$ and $n_\gamma(i)$ denote when the cell appears in the characterization vehicle's path and in the search vehicle's path, respectively. For example, if cell 5 is the 4th cell the characterization vehicle visits and the 2nd cell the search vehicle

visits, then $n_\eta(i=5)=4$ and $n_\gamma(i=5)=2$. We note that when $n_\eta(i) > n_\gamma(i)$, there is no gain of characterizing the cell since the increase in the uncertainty of the environment will not affect the search results. Thus, the modification yields

$$\mathbf{I}\big(i, \gamma(y_i), \gamma, \eta\big) = \begin{cases} 1 & i \in \gamma(y_i) \text{ and } n_\eta(i) \leq n_\gamma(i) \\ 0 & \text{otherwise} \end{cases} \tag{3.20}$$

3.4 Sensors operate on the same vehicle

We lastly consider the case that a single vehicle is equipped with an environmental characterization sensor and a search sensor, and that both sensors can operate simultaneously. Thus, when the vehicle visits a location, it acquires both a search measurement $z \in Z$ and an environment measurement $y \in Y$ at the same time. This scenario extends our findings in Chapter 2 where the vehicle was assumed to acquire only search measurements. We derive the decision-theoretic cost function that accounts for the environment measurement y for both cases: 1) when the search objective is to maximize the estimation accuracy, and 2) when the search objective is to minimize the risk of incorrect estimations.

3.4.1 Search objective: maximize estimation accuracy

Consider the case where the search objective is to maximize estimation accuracy, let $w_1, w_2, \ldots, w_m$ be a set of environments, and let $W\big(w_j\big)$ denote estimation accuracy conditioned on the environment w_j.

$$W\big(w_j\big) = \max_x P\big(X = x \mid z, w_j\big) \tag{3.21}$$

We say $w_i \preceq w_j$ if and only if $W\big(w_i\big) \leq W\big(w_j\big)$. Note $W\big(w_j\big)$ in (3.21) is the accuracy of the estimate of the number of objects at a location while $V\big(w_j\big)$ in (3.9) is the expected accuracy when a measurement z has not yet been acquired. Suppose the environments $w_1, \ldots, w_m$ are distinct and *ordered* (as defined in Section 3.3.2) so that $w_1 \prec w_2 \prec \ldots \prec w_m$, let e be the true environment in a cell, and let $\delta_E(y)$ be an estimate of the environment based on the environment measurement y. When the true environment is e, the loss due to the

estimate $\delta_E(y)$ is defined

$$L_W\big(e, \delta_E(y)\big) = \begin{cases} c_1\big(W\big(\delta_E(y)\big) - W\big(e\big)\big) & \text{if } \delta_E(y) \succ e \\ c_2\big(W\big(e\big) - W\big(\delta_E(y)\big)\big) & \text{if } \delta_E(y) \preceq e \end{cases} \tag{3.22}$$

where $c_1, c_2 > 0$ are again the relative costs of over and underestimation. Then, the posterior expected loss of computing the environment estimate $\delta_E(y)$, and the corresponding Bayes estimator $\delta_E^\star(y)$ are

$$\mathbb{E}\Big[L_W\big(e, \delta_E(y)\big) \mid z, y\Big] = \sum_{j=1}^{m} P\big(w_j \mid y\big) L_W\big(w_j, \delta_E(y)\big) \tag{3.23}$$

$$\delta_E^\star(y) = \arg\min_{\delta_E(y)} \mathbb{E}\Big[L_W\big(e, \delta_E(y)\big) \mid z, y\Big] \tag{3.24}$$

Given measurement z and the estimate $\delta_E^\star(y) \in w_1, w_2, \ldots, w_m$ from measurement y, the probability that the estimate of the number of objects at a location is correct is computed from

$$\mathbb{E}\Big[U\big(x, \delta_X(z)\big) \mid z, \delta_E^\star(y)\Big] = \max_x P\big(X = x \mid z, \delta_E^\star(y)\big) \tag{3.25}$$

In order to assess the benefit of visiting a location, we compute the *estimated* estimation accuracy in (3.25) for each possible set of observations $z \in Z,\ y \in Y$. Then, the expected estimation accuracy before visiting a location can be computed

$$\mathbb{E}\Big[W\big(\delta_E(y)\big)\Big] = \sum_z \sum_y P\big(z, y\big)\ \max_x P\big(X = x \mid z, \delta_E^\star(y)\big) \tag{3.26}$$

where

$$P\big(z, y\big) = \sum_x \sum_{w_j} P\big(z \mid x, w_j\big) P\big(y \mid w_j\big) P\big(x\big) P\big(w_j\big) \tag{3.27}$$

We again consider the candidate search path γ, the finite collection of feasible search paths Ω_γ, and the budget constraint $\bar{\mathcal{C}}_\gamma$ on the vehicle. Let y_{q_i} be the set of independent environment measurements acquired at q_ith cell. The expected estimation accuracy for traversing γ is

$$\mathbb{E}\Big[W\big(\delta_E(y_\gamma)\big)\Big] = \prod_{q_i \in \gamma} \mathbb{E}\Big[W\big(\delta_E(y_\gamma)\big)\Big] \times \prod_{i \in S \setminus \gamma} \max_{x_i} P\big(x_i\big) \tag{3.28}$$

and the optimal path is

$$\gamma^\star = \arg\max_{\gamma \in \Omega_\gamma} \mathbb{E}\Big[W\big(\delta_E(y_\gamma)\big)\Big] \tag{3.29}$$

subject to

$$\mathcal{C}(\gamma) \leq \bar{\mathcal{C}}_\gamma \tag{3.30}$$

3.4.2 Search objective: minimize the risk associated with incorrect estimations

We now consider that the search objective is to minimize the risk associated with incorrectly estimating the number of targets, and we denote by $W\big(w_j\big)$ the conditional posterior expected loss in (2.19)

$$W\big(w_j\big) = \sum_x P\big(X = x \mid z, w_j\big) L_X\big(x, \delta_X(z)\big) \tag{3.31}$$

where the loss function $L_X\big(x, \delta_X(z)\big)$ is defined in (2.18). The corresponding loss function $L_W\big(e, \delta_E(y)\big)$, the posterior expected loss of computing the environment estimate $\mathbb{E}\big[L_W\big(e, \delta_E(y)\big) \mid z, y\big]$, and the Bayes estimator $\delta_E^\star(y)$ follows from (3.22), (3.23), and (3.24), respectively.

Given $P\big(z, y\big)$, the probability of acquiring a particular search measurement z and a particular environment measurement y in (3.27), the value of searching a location can be computed from

$$\mathbb{E}\Big[W\big(\delta_E(y)\big)\Big] = \sum_z \sum_y P(z,y) \sum_x P\big(X = x \mid z, w_j\big) L_X\big(x, \delta_X(z)\big) \tag{3.32}$$

The value of traversing a path $\gamma = [q_1, q_2, \ldots, q_N]$ is the sum over the value of searching each location q_i along the path

$$\mathbb{E}\Big[W\big(\delta_E(y_\gamma)\big)\Big] = \sum_{q_i \in \gamma} \mathbb{E}\Big[W\big(\delta_E(y_\gamma)\big)\Big] \tag{3.33}$$

and, the corresponding optimal path subject to the budget constraint in (3.30) is

$$\gamma^\star = \arg\max_{\gamma \in \Omega_\gamma} \mathbb{E}\Big[W\big(\delta_E(y_\gamma)\big)\Big]. \tag{3.34}$$

3.5 Computing near-optimal paths for environmental characterization vehicle

In this section, we show two approximation methods to compute the near-optimal paths for the environmental characterization vehicle. The first method aims to approximate the gain of characterizing a path, and the second method aims to approximate the gain of characterizing a cell. While the first method achieves a provably near-optimal performance, the second method scales better with a larger planning horizon and with a larger search area.

3.5.1 Approximating the characterization gain of a path

Computing the optimal path for the environment characterization vehicle in (3.18) can be computationally very expensive. This is mainly due to large computational requirements of computing the optimal search path for each set of environment measurements along a candidate path η. Thus, in this section, we propose an approximate method that reduces the computational complexity of the solution in (3.18). Our approximate solution yields provably near-optimal paths.

Let $\bar{\Omega}_\eta$ be the set of environment characterization paths such that $\mathcal{C}(\eta) \leq \bar{\mathcal{C}}_\eta$, let $|\cdot|$

denote the cardinality of a set (or an array), and let $\mathcal{S}$ denote the computational complexity of computing the optimal search paths. Then, the solution in (3.18) has a computational complexity of $\mathcal{O}\big(|\bar{\Omega}_\eta| m^{|\eta|} \mathcal{S}\big)$ where m is the number of environments in the search domain. This shows that the exponential increase in the computational complexity is dominated by the large planning horizon for the characterization vehicle. One approach to reduce this computational complexity is to use a receding horizon strategy where we compute the paths for a shorter horizon. While a receding horizon approach may require less computational power compared to computing the paths for the entire planning horizon, it may still be infeasible unless the considered planning horizon is sufficiently small. Instead, our approach to reduce the complexity of the solution in (3.18) aims to approximate the characterization gain of traversing a path.

We start with re-arranging the terms in (3.17) by partitioning a path η into two parts: a cell $q_i \in \eta$ and the other cells in the path

$$\begin{aligned}\mathbb{E}\big[G_\eta\big] &= \sum_{y_\eta} P\big(Y_\eta = y_\eta\big) \sum_{q_i \in \eta} \mathbf{I}\big(q_i, \gamma^\star(y_\eta)\big)\ \mathcal{R}(y_{q_i}) \\ &= \sum_{q_i \in \eta} \sum_{y_{q_i}}\ \mathcal{R}(y_{q_i}) \sum_{y_{\eta \setminus q_i}} \mathbf{I}\big(q_i, \gamma^\star(y_\eta)\big)\ P\big(Y_\eta = y_\eta\big) \end{aligned} \tag{3.35}$$

where $\eta \setminus q_i$ denotes the set of cells in η except cell q_i. The terms up to the third summation in (3.35) denote the characterization gain of acquiring the environment measurement y from cell $q_i \in \eta$, and the other terms that start with the third summation denote how likely it is that sampling cell q_i will improve the performance of a follow-on search mission. That is, it represents the chance that cell q_i will be visited during a follow-on search mission based on the environment measurements that we may acquire along the path η. Note that since $\eta = \{q_1, q_2, \ldots, q_M\}$, the joint probability $P\big(Y_\eta = y_\eta\big)$ in (3.35) can be expressed

$$P\big(Y_\eta = y_\eta\big) = P\big(Y_{q_1} = y_{q_1}\big) \times \cdots \times P\big(Y_{q_M} = y_{q_M}\big) \tag{3.36}$$

Thus, we can re-write (3.35)

$$\mathbb{E}\big[G_\eta\big] = \sum_{q_i \in \eta} \sum_{y_{q_i}} \Big(P\big(Y_{q_i} = y_{q_i}\big) \mathcal{R}(y_{q_i}) \Big) \times \bar{\mathbf{P}}_{\eta \setminus q_i} \tag{3.37}$$

where $\bar{\mathbf{P}}_{\eta \setminus q_i}$ is the total probability of every possible sets of the environment measurements $y_{q_1}, y_{q_2}, \ldots, y_{q_{i-1}}, y_{q_{i+1}}, \ldots, y_{q_M}$ such that the optimal search path associated with the updated environment distributions visits cell q_i

$$\begin{aligned} \bar{\mathbf{P}}_{\eta \setminus q_i} &= \sum_{y_{\eta \setminus q_i}} \mathbf{I}\big(q_i, \gamma^\star(y_\eta)\big) \; P\big(Y_{\eta \setminus q_i} = y_{\eta \setminus q_i}\big) \\ &= \sum_{y_{\eta \setminus q_i} \colon \; q_i \in \gamma^\star(y_\eta)} P\big(Y_{\eta \setminus q_i} = y_{\eta \setminus q_i}\big) \end{aligned} \tag{3.38}$$

Indeed, the computational cost of (3.37) is dominated by $\bar{\mathbf{P}}_{\eta \setminus q_i}$ in (3.38). Thus, we use a sample-based method to compute an empirical estimate of $\bar{\mathbf{P}}_{\eta \setminus q_i}$, which results in a significant speed-up in computing the characterization path. For each cell $q_i \in \eta$ and for every environment measurement $y_{q_i} \in Y$, we perform $\bar{N}$ trials where, in each trial, we randomly sample an environment measurement from the probability distribution $P\big(Y_{q_j} = y_{q_j}\big)$ for all $q_j \in \eta$ such that $j \neq i$. Then, with the updated environment distributions we compute the optimal search path and assess if it visits cell q_i. We simply count the number of times cell q_i is visited by the resulting search path out of $\bar{N}$ trials, and we denote this number by k. Since this is repeated for every other cell in the path, we may sample the same environment measurement y_{q_i} from cell q_i during a trial of another cell. Let $\bar{N}_{y_{q_i}}$ be the number of times y_{q_i} is sampled in cell q_i during the trials of the other cells in path η and let $k_{y_{q_i}}$ be the number of times cell q_i is contained in the corresponding search path out of these $\bar{N}_{y_{q_i}}$ trials. Then, the empirical estimate of $\bar{\mathbf{P}}_{\eta \setminus q_i}$ is

$$\hat{\mathbf{P}}_{\eta \setminus q_i} = \frac{k + k_{y_{q_i}}}{\bar{N} + \bar{N}_{y_{q_i}}} \tag{3.39}$$

Obtaining a close estimate of $\bar{\mathbf{P}}_{\eta \setminus q_i}$ is important to compute a near-optimal characterization path. We show that a bound on the distance between $\bar{\mathbf{P}}_{\eta \setminus q_i}$ and $\hat{\mathbf{P}}_{\eta \setminus q_i}$ can be computed.

We first note that after each trial for a cell, that cell is either contained in the follow-on search path, or it is not contained. Thus, we can cast each trial as a Bernoulli trial where the result of the trial is either 1 if the cell is contained in the search path, or it is 0 if the cell is not contained. Then, we use Hoeffding's inequality [56] to obtain a probabilistic bound on the difference between $\bar{\mathbf{P}}_{\eta\setminus q_i}$ and $\hat{\mathbf{P}}_{\eta\setminus q_i}$

$$P\big(|\bar{\mathbf{P}}_{\eta\setminus q_i} - \hat{\mathbf{P}}_{\eta\setminus q_i}| < \epsilon\bar{\mathbf{N}}\big) \geq 1 - 2\exp^{-2\epsilon^2\bar{\mathbf{N}}} \tag{3.40}$$

where $\bar{\mathbf{N}} = \bar{N}+\bar{N}_{y_{q_i}}$ and $\epsilon > 0$. Replacing $\bar{\mathbf{P}}_{\eta\setminus q_i}$ with its estimate $\hat{\mathbf{P}}_{\eta\setminus q_i}$ in (3.37) approximates the characterization gain of a path. We denote the approximate characterization gain of a path η by $\mathbb{E}\big[\hat{G}_\eta\big]$

$$\mathbb{E}\big[\hat{G}_\eta\big] = \sum_{q_i\in\eta}\sum_{y_{q_i}} \frac{k + k_{y_{q_i}}}{\bar{N} + \bar{N}_{y_{q_i}}} P\big(Y_{q_i} = y_{q_i}\big)\mathcal{R}(y_{q_i}) \tag{3.41}$$

Finally, we select the path that maximizes the approximate characterization gain in (3.41) subject to the budget constraint in (3.19).

$$\hat{\eta} = \arg\max_{\eta\in\Omega_\eta} \mathbb{E}\big[\hat{G}_\eta\big] \tag{3.42}$$

Now, we define the following theorem and the corollary, where we first bound the difference between the characterization gain and the approximate characterization gain for a path, and we then bound the difference between the optimal characterization gain and the approximately optimal characterization gain.

Theorem 1. The probability of the difference between the characterization gain (3.37) and its estimate (3.41) for a path η satisfying a specific bound is expressed

$$P\Big(\big|\mathbb{E}\big[G_\eta\big] - \mathbb{E}\big[\hat{G}_\eta\big]\big| < \epsilon\bar{\mathbf{N}}|\eta|\Big) \geq 1 - 2\exp^{-2\epsilon^2\bar{\mathbf{N}}} \tag{3.43}$$

Proof. First, observe that when $c_1, c_2 \leq 1$ in (3.11)

$$\begin{aligned}\mathcal{R}(y) &= \mathbb{E}\Big[L_V\big(e,\delta_E^\star\big)\Big] - \mathbb{E}\Big[L_V\big(e,\delta_E^\star(y)\big) \mid y \in Y\Big] \\ &\leq \mathbb{E}\Big[L_V\big(e,\delta_E^\star\big)\Big] \\ &\leq \max_{w_i,w_j}\big(V\big(w_i\big) - V\big(w_j\big)\big) \\ &\leq 1\end{aligned}$$

for all $y \in Y$.

The difference between $\mathbb{E}\big[G_\eta\big]$ and $\mathbb{E}\big[\hat{G}_\eta\big]$ is

$$\begin{aligned}\big|\mathbb{E}\big[G_\eta\big] - \mathbb{E}\big[\hat{G}_\eta\big]\big| &= \Big|\sum_{q_i\in\eta}\sum_{y_{q_i}}\Big(P\big(Y_{q_i}=y_{q_i}\big)\mathcal{R}(y_{q_i})\Big)\times\bar{\mathbf{P}}_{\eta\backslash q_i} \\ &\qquad - \sum_{q_i\in\eta}\sum_{y_{q_i}}\Big(P\big(Y_{q_i}=y_{q_i}\big)\mathcal{R}(y_{q_i})\Big)\times\hat{\mathbf{P}}_{\eta\backslash q_i}\Big| \\ &= \Big|\sum_{q_i\in\eta}\sum_{y_{q_i}}\Big(P\big(Y_{q_i}=y_{q_i}\big)\mathcal{R}(y_{q_i})\Big)\times\big(\bar{\mathbf{P}}_{\eta\backslash q_i} - \hat{\mathbf{P}}_{\eta\backslash q_i}\big)\Big| \\ &\leq \sum_{q_i\in\eta}\sum_{y_{q_i}}\Big(P\big(Y_{q_i}=y_{q_i}\big)\mathcal{R}(y_{q_i})\Big)\times\big|\bar{\mathbf{P}}_{\eta\backslash q_i} - \hat{\mathbf{P}}_{\eta\backslash q_i}\big| \\ &\leq \sum_{q_i\in\eta}\sum_{y_{q_i}}P\big(Y_{q_i}=y_{q_i}\big)\times\big|\bar{\mathbf{P}}_{\eta\backslash q_i} - \hat{\mathbf{P}}_{\eta\backslash q_i}\big| \\ &= \sum_{q_i\in\eta}\big|\bar{\mathbf{P}}_{\eta\backslash q_i} - \hat{\mathbf{P}}_{\eta\backslash q_i}\big| \\ &= |\eta|\big|\bar{\mathbf{P}}_{\eta\backslash q_i} - \hat{\mathbf{P}}_{\eta\backslash q_i}\big|\end{aligned}$$

Due to the bound on the difference between $\bar{\mathbf{P}}_{\eta\backslash q_i}$ and $\hat{\mathbf{P}}_{\eta\backslash q_i}$ in (3.38), it follows that

$$\begin{aligned}P\big(\big|\mathbb{E}\big[G_\eta\big] - \mathbb{E}\big[\hat{G}_\eta\big]\big| < \epsilon\bar{\mathbf{N}}|\eta|\big) &= P\big(|\eta||\bar{\mathbf{P}}_{\eta\backslash q_i} - \hat{\mathbf{P}}_{\eta\backslash q_i}| < \epsilon\bar{\mathbf{N}}|\eta|\big) \\ &= P\big(|\bar{\mathbf{P}}_{\eta\backslash q_i} - \hat{\mathbf{P}}_{\eta\backslash q_i}| < \epsilon\bar{\mathbf{N}}\big) \\ &\geq 1 - 2\exp^{-2\epsilon^2\bar{\mathbf{N}}}\end{aligned}$$

□

Corollary 1. For the optimal characterization path $\eta^\star$ in (3.18) and the approximate path $\hat{\eta}$ in (3.42), the difference in expected characterization gain satisfies

$$P\big(\mathbb{E}\big[G_{\eta^\star}\big] - \mathbb{E}\big[G_{\hat{\eta}}\big] < 2\epsilon\bar{\mathbf{N}}|\eta|\big) \geq 1 - 2\exp^{-2\epsilon^2\bar{\mathbf{N}}} \tag{3.44}$$

Proof. By Theorem 1, we obtain the bounds for $\eta^\star$ in (3.18) and for $\hat{\eta}$ in (3.42)

$$\begin{aligned} P\big(\big|\mathbb{E}\big[G_{\eta^\star}\big] - \mathbb{E}\big[\hat{G}_{\eta^\star}\big]\big| < \epsilon\bar{\mathbf{N}}|\eta^\star|\big) &\geq 1 - 2\exp^{-2\epsilon^2\bar{\mathbf{N}}} \\ P\big(\big|\mathbb{E}\big[G_{\hat{\eta}}\big] - \mathbb{E}\big[\hat{G}_{\hat{\eta}}\big]\big| < \epsilon\bar{\mathbf{N}}|\hat{\eta}|\big) &\geq 1 - 2\exp^{-2\epsilon^2\bar{\mathbf{N}}} \end{aligned}$$

Note that since $\hat{\eta}$ maximizes (3.42), $\mathbb{E}\big[\hat{G}_{\hat{\eta}}\big] \geq \mathbb{E}\big[\hat{G}_{\eta^\star}\big]$. Hence,

$$\begin{aligned} \mathbb{E}\big[G_{\eta^\star}\big] - \mathbb{E}\big[G_{\hat{\eta}}\big] &= \mathbb{E}\big[G_{\eta^\star}\big] - \mathbb{E}\big[\hat{G}_{\eta^\star}\big] + \mathbb{E}\big[\hat{G}_{\eta^\star}\big] - \mathbb{E}\big[G_{\hat{\eta}}\big] \\ &\leq \big(\mathbb{E}\big[G_{\eta^\star}\big] - \mathbb{E}\big[\hat{G}_{\eta^\star}\big]\big) + \big(\mathbb{E}\big[\hat{G}_{\hat{\eta}}\big] - \mathbb{E}\big[G_{\hat{\eta}}\big]\big) \\ &\leq \big|\mathbb{E}\big[G_{\eta^\star}\big] - \mathbb{E}\big[\hat{G}_{\eta^\star}\big]\big| + \big|\mathbb{E}\big[\hat{G}_{\hat{\eta}}\big] - \mathbb{E}\big[G_{\hat{\eta}}\big]\big| \end{aligned}$$

Then, assuming $|\eta^\star| = |\hat{\eta}| = |\eta|$, the error in approximation of the optimal characterization gain is

$$\begin{aligned} P\big(\mathbb{E}\big[G_{\eta^\star}\big] - \mathbb{E}\big[G_{\hat{\eta}}\big] < 2\epsilon\bar{\mathbf{N}}|\eta|\big) \geq \\ P\big(\big|\mathbb{E}\big[G_{\eta^\star}\big] - \mathbb{E}\big[\hat{G}_{\eta^\star}\big]\big| + \big|\mathbb{E}\big[\hat{G}_{\hat{\eta}}\big] - \mathbb{E}\big[G_{\hat{\eta}}\big]\big| < 2\epsilon\bar{\mathbf{N}}|\eta|\big) \\ \geq 1 - 2\exp^{-2\epsilon^2\bar{\mathbf{N}}} \end{aligned}$$

□

The proposed approximation approach yields a provably near-optimal path for the envi-

ronment characterization vehicle. The computational complexity of the solution is reduced from $\mathcal{O}\big(|\bar{\Omega}_\eta| m^{|\eta|} \mathcal{S}\big)$ to $\mathcal{O}\big(|\bar{\Omega}_\eta||\eta| m \bar{N} \mathcal{S}\big)$. In general, choosing a larger value for $\bar{N}$ is likely to reduce the approximation error. However, our preliminary results show that a small value for $\bar{N}$ is often sufficient to obtain a close approximation.

3.5.2 Approximating the characterization gain of a cell

In this section, we introduce an alternative approach to approximate the optimal characterization gain. Our alternative approach assumes that the search paths are composed of sequences of parallel straight lines. This assumption arises often in subsea applications that rely on side-scan imaging sonar (see, for example, [41, 42]). Unlike the proposed approach in Section 3.5.1, our approach in this section can only apply to certain classes of mapping problems.

Suppose the search area $\mathcal{G}$ consists of a set of parallel straight lines (a line is either a row or a column) $l_1, l_2, \ldots, l_{n_l}$. When each line corresponds to a row, $n_l = n_r$, and when each line corresponds to a column, $n_l = n_c$. Suppose that the jth line traverses the cells $q_{j1}, q_{j2}, \ldots, q_{jk}$. Thus, when the vehicle traverses line l_j, it samples from cells $q_{j1}, q_{j2}, \ldots, q_{jk}$. Let η be a candidate path for the environment characterization vehicle that consists of lines $l_1, l_2, \ldots l_k$, and consider that cell $i \in \eta$ is contained in l_j. We claim that the value of characterizing cell i along path η can be closely approximated by the value of characterizing cell i along line l_j. That is

$$\mathbb{E}\Big[G_{q_i \in \eta}\Big] = \sum_{y_{q_i}} \Big(P\big(Y_{q_i} = y_{q_i}\big) \mathcal{R}(y_{q_i}) \Big) \times \bar{\mathbf{P}}_{\eta \setminus q_i} \tag{3.45}$$

$$\approx \sum_{y_{q_i}} \Big(P\big(Y_{q_i} = y_{q_i}\big) \mathcal{R}(y_{q_i}) \Big) \times \bar{\mathbf{P}}_{l_j \setminus q_i} \tag{3.46}$$

where

$$\bar{\mathbf{P}}_{l_j \setminus q_i} = \sum_{y_{l_j \setminus q_i} :\ q_i \in \gamma^\star(y_{l_j})} P\big(Y_{l_j \setminus q_i} = y_{l_j \setminus q_i}\big) \tag{3.47}$$

Due to (3.46), we can compute the value of characterizing a particular cell by only looking at the cells in the associated line (a row or a column). By doing so, we can compute the characterization gain for each cell individually. Then, the characterization gain of a path is simply the sum of the characterization gains of each cell in that path. We note that we do not have a formal guarantee of how closely (3.46) approximates (3.45). However, our initial tests as well as intuition suggest that (3.45) can be well-approximated by (3.46).

Since computing $\bar{\mathbf{P}}_{l_j \setminus q_i}$ in (3.47) can be very expensive when the length of a line is large, we instead compute an empirical estimate of $\bar{\mathbf{P}}_{l_j \setminus q_i}$ as described in Section 3.5.1. For each cell in the search grid, we perform $\bar{N}$ trials to compute an empirical estimate of $\bar{\mathbf{P}}_{l_j \setminus q_i}$, where in each trial we sample environment measurements from the remaining cells in line l_j. Then, similar to Section 3.5.1, the empirical estimate of $\bar{\mathbf{P}}_{l_j \setminus q_i}$ follows from (3.39)

$$\hat{\mathbf{P}}_{l_j \setminus q_i} = \frac{k + k_{y_{q_i}}}{\bar{N} + \bar{N}_{y_{q_i}}} \tag{3.48}$$

the characterization gain for traversing a candidate path η is

$$\mathbb{E}\big[\hat{G}_\eta\big] = \sum_{q_i \in \eta} \sum_{y_{q_i}} \frac{k + k_{y_{q_i}}}{\bar{N} + \bar{N}_{y_{q_i}}} P\big(Y_{q_i} = y_{q_i}\big) \mathcal{R}(y_{q_i}) \tag{3.49}$$

and, the approximate characterization gain subject to the budget constraint in (3.19) is

$$\hat{\eta} = \arg\max_{\eta \in \Omega_\eta} \mathbb{E}\big[\hat{G}_\eta\big] \tag{3.50}$$

We note that (3.49) and (3.41) are not the same. In (3.41), the value of characterizing a cell q_i in a path η is computed with respect to η. However, in (3.49), the gain of characterizing each cell is computed individually, and thus, the gain of characterizing the path η is simply the sum over the cell-wise characterization gains along the path. In addition, this approximation approach allows us to use the well-known approximate solvers (e.g., branch-and-bound planner and Monte Carlo tree search planner shown in Chapter 6) that are commonly employed in similar problems. Thus, the computational complexity of the solution is reduced from $\mathcal{O}\big(|\bar{\Omega}_\eta| m^{|\eta|} \mathcal{S}\big)$ to $\mathcal{O}\big((rm\bar{N} + 1)\mathcal{S}\big)$.

3.6 Numerical results

In this section, we present simulation results that show the efficacy of the proposed search and environmental characterization strategies. We present numerical illustrations for two scenarios. In one case, search and environmental characterization sensors are on different vehicles and environmental characterization is performed prior to search. In the other case, search and environmental characterization sensors are on the same vehicle and both activities occur simultaneously. For both cases, we assume that the search objective is to maximize the probability that our estimates of the number of targets is correct. Thus, when the sensors operate on separate vehicles, the objective of the search vehicle is to maximize anticipated estimation accuracy in (2.15), and the objective of the characterization vehicle is to maximize the expected gain of characterization in (3.17). When both sensors operate on the same vehicle, the objective of the vehicle is to maximize expected estimation accuracy in (3.28).

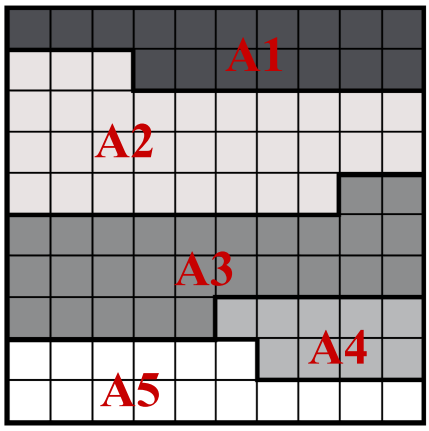
The search environment characteristics as well as the assumptions on the search vehicle given in Section 2.3 also apply to the characterization mission and the environmental characterization vehicle. We consider that the sensor model for environment characterization is

$$a_{ij} = P\big(Y = w_i \mid E = w_j\big) \quad \text{for all} \quad i, j \in \{1, 2, 3\} \tag{3.51}$$

where a_{ii} is the probability of observing the true environment w_i. For the numerical illustrations, we use the characterization sensor model with $a_{11} = 0.9$, $a_{22} = 0.92$, $a_{33} = 0.94$. That is, for example, there is 0.9 probability of acquiring environment measurement w_1 when w_1 is the true environment at the location. The noisy environment observations are due to nonzero probabilities of observing environment w_i when true environment is w_j, denoted by a_{ij} for $i \neq j$. We assume the probability of acquiring incorrect environment measurement is the same for all possible environments other than the true environment. For example, when the true environment at a location is w_1, since $a_{11} = 0.9$, the probability of acquiring environment measurement w_2 and probability of acquiring environment measurement w_3 are $a_{21} = a_{31} = 0.05$. The probability of detection, D, and the probability of at least one false alarm, F, for each environment are shown in Table 2.1.

Fig. 4.1 shows a search area that is partitioned into regions A1 through A5. For each

region, the corresponding probability distribution $[p_1, p_2, p_3]$ is given, where p_j is the probability that the environment is w_j. The relative costs of over and underestimating the environmental conditions are $c_1 = 1$ and $c_2 = 3$ so that overestimation is penalized more than underestimation. The mission length is 60 for the search and search/environmental characterization vehicles and 35 for the characterization vehicle.



Region	Probability distribution		
	w_1	w_2	w_3
A1	[1.00	0.00	0.00]
A2	[0.10	0.25	0.65]
A3	[0.40	0.30	0.40]
A4	[0.15	0.80	0.05]
A5	[0.05	0.05	0.90]

Figure 3.1: Search area and cell-wise environment distributions

We define the *error in search performance* after a mission as the difference between the actual estimation accuracy when the true environment is known and the anticipated estimation accuracy when the environment is uncertain. We use the error in search performance as a measure to evaluate the efficacy of the proposed approaches in each scenario, and show that the proposed approach yields smaller search performance error, which is predicted by our selection of cost function. We also show that search performance (probability of correct estimate) increases modestly, although our approach does not directly seek to increase estimation accuracy.

When the sensors are on separate vehicles and characterization precedes search, we compare the proposed approach in (3.17) with the entropy change maximization method described in Section 3.3.1. Fig. 3.2e shows the trajectory for the environmental characterization vehicle when using our proposed approach in (3.17), which seeks to characterize the environment in locations that are expected to yield the greatest reduction of uncertainty in anticipated estimation accuracy. In contrast, Figure 3.2f shows the path of an environmental

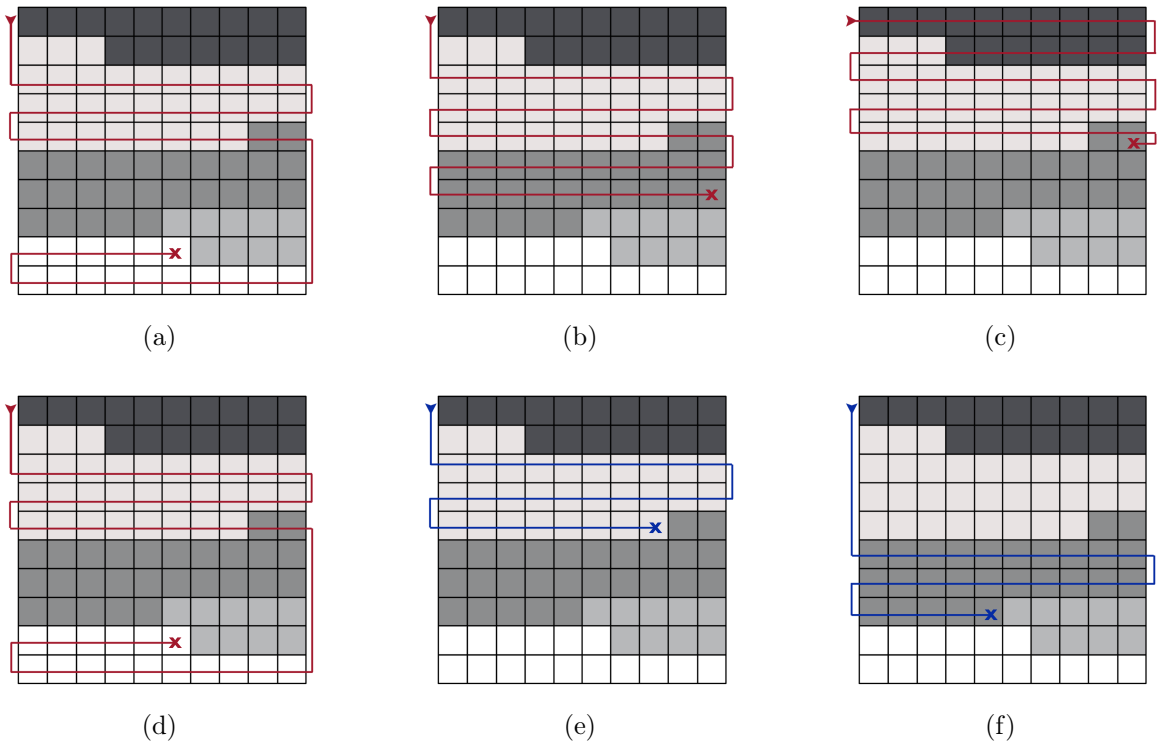


Figure 3.2: Optimal trajectories for search and characterization. Figures (a-c): trajectories for the case both sensors operate on the same vehicle when (a) proposed approach is employed (b) entropy change maximization method is employed, and (c) the mowing-the-lawn approach is employed. Figures (e-f): characterization vehicle's trajectories for the case the sensors operate on separate vehicles when the characterization locations are selected (e) by our proposed approach, (f) by entropy change maximization method. Figure (d) shows the search vehicle's trajectory when no environment information is acquired.

characterization vehicle when the path is selected by maximizing the change in entropy of the environmental distributions. Neither environmental characterization path visits A1 because the environments in those locations are completely known. We note that the environmental characterization path in Figure 3.2e that was selected using our approach does not visit the most uncertain environments. We find in practice that it tends to visit environments that are both uncertain *and* likely to be where follow-on search missions will occur.

When both sensors operate on the same vehicle, we compare the proposed approach in (3.29) with the entropy change maximization method and with a mowing-the-lawn approach. The latter arises often in subsea applications such as mine-hunting. We note that

the entropy change maximization method described in Section 3.3.1 accounts only for the entropy change of the environmental distributions. However, when both sensors are placed on the same vehicle, the vehicle acquires environmental measurements and search measurements simultaneously. Thus, we modify (3.8) as

$$\gamma^\star = \arg\max_{\gamma\in\Omega_\gamma} \big(J_\gamma(X) + \psi J_\gamma(E)\big)$$

where $J(X)$ denotes the entropy change in X, the number of targets, and ψ is the relative weight of the entropy change in E compared to the entropy change in X. Since the objective is to reduce the uncertainty in the number of targets, we choose $0 < \psi < 1$. Fig. 3.2c shows the mowing-the-lawn trajectory where the vehicle travels through the search area back and forth without planning the path until the mission length is met. Fig. 3.2a shows the trajectory for the proposed approach and Fig. 3.2b shows the trajectory for the entropy change maximization method with $\psi = 0.5$. We also compute the optimal search trajectory when there is no environmental characterization to show the value of acquiring environmental information. The corresponding trajectory for this case is shown in Fig. 3.2d.

We expect that for both scenarios our proposed approach yields better search performance compared to the other path planning strategies in Fig. 3.2. That is, when search and environmental characterization missions are performed on the same vehicle, if the search locations are selected using our approach as in Fig. 3.2a, the search performance is expected to be better compared to selecting the locations using entropy change maximization as in Fig. 3.2b or mowing the lawn as in Fig. 3.2c. When the sensors operate on separate vehicles, we expect that selecting the characterization locations using our approach as in Fig. 3.2e will yield greater improvement in the performance of a follow-on search mission compared to selecting the locations using entropy change maximization as in Fig. 3.2f.

Search performance after a mission depends on the observations acquired during the mission. Thus, we conduct Monte Carlo simulations to assess the effects due to the random nature of observations. For each cell in the search area, we randomly generate the true environment e from the environmental distributions in Fig. 4.1 and the true number of targets x from a uniform distribution. Assuming that a cell can be visited by a vehicle at most k

times, we randomly generate the set of search measurements z and the set of environmental measurements y from the sensor models $P(z \mid x, e)$ and $P(y \mid e)$ given the true environment e and the true number of targets x. When a vehicle visits a location, it acquires randomly generated observation(s). For each test, we compute the anticipated search performance and the actual search performance. Note that the actual search performance can be computed since the true environment is assumed to be known. We then compute the error in search performance which is the difference between the anticipated search performance and the actual search performance. We show that the error in search performance is significantly reduced when our proposed approach is employed.

3.6.1 Both sensors operating simultaneously on a single vehicle

Fig. 3.3 shows the results after 10,000 iterations for the case both sensors operate on the same vehicle. Fig. 3.3a on the left is the percentage of occurrences of the error in search performance, and Fig. 3.3b on the right is the percentage of occurrences of the actual search performance. For convenience, we compute the actual search performance after traversing an optimal path γ and acquiring the search measurements z_γ as

$$-\Big(\log\big(\prod_{i\in\mathcal{G}}\max_{x_i}P(x_i)\big)-\log\big(\prod_{i\in\gamma}\max_{x_i}P(x_i\mid z_i,e_i)\times\prod_{i\in\mathcal{G}\setminus\gamma}\max_{x_i}P(x_i)\big)\Big)$$

where e_i is the actual environment in cell i. That is, the actual search performance is the difference between the prior certainty in the number of targets before acquiring any measurement and the posterior certainty in the number of targets after acquiring the search measurements along the path. Loosely speaking, the actual search performance plotted in Fig. 3.3b represents the amount of information we acquire of the number of targets after traversing the corresponding optimal search path. Thus, smaller values for Fig. 3.3a imply less error in search performance and larger values for Fig. 3.3b imply better search performance. The displayed results are the negative log of the computed search performance. The subplots from top to bottom are the results when 1) our proposed approach is employed, 2) the entropy change maximization method is employed, 3) the mowing-the-lawn approach

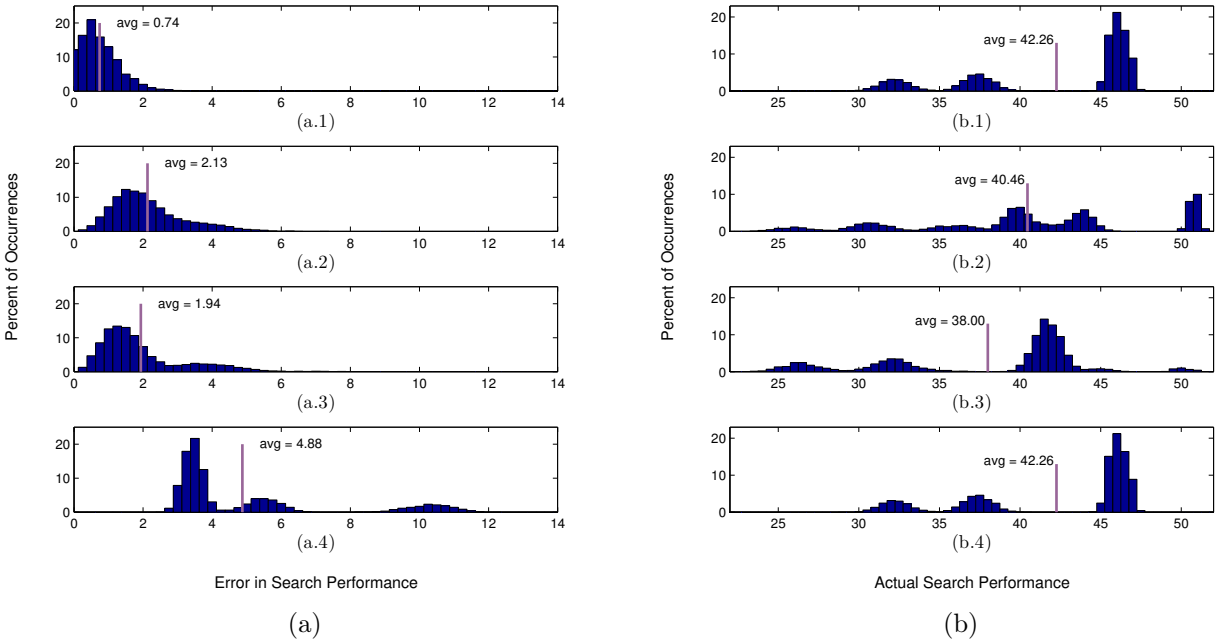


Figure 3.3: Percentage of occurrences for (a) error in search performance and (b) actual search performance when both sensors operate on *the same vehicle*. From top to bottom, (a.1) and (b.1) correspond to the proposed approach, (a.2) and (b.2) correspond to the entropy change maximization method, (a.3) and (b.3) correspond to the mowing-the-lawn approach, and (a.4) and (b.4) correspond to the case where environment information is not available. Note that the horizontal axis is the negative log of the results. Smaller values for (a) imply less error in search performance and larger values for (b) imply better search performance.

is employed, and 4) environmental information is not available so that the vehicle acquires only the search measurements. The average value of results for each test is also shown in the plots. The simulations show that

- The proposed approach yields smaller error in search performance compared to the entropy change maximization and mowing-the-lawn. With respect to the case where there is no environment information (in Fig. 3.3a.4), our proposed approach achieves 85% error reduction while entropy change maximization achieves 56% and mowing-the-lawn achieves 60%, on average. In addition, the actual search performance when using our approach is no worse than the actual search performance when using the other methods.

- Fig. 3.3a.1 shows that in many of the iterations, the error in search performance is very close to zero. This implies that, in these trials, we correctly estimate the environmental conditions in each visited cell. We note that this is also due to the sensor model we choose in (3.51) for environment characterization. Choosing a sensor model with greater probabilities of observing the true environments will further reduce the error while choosing a sensor model with smaller probabilities of observing the true environments will increase the error. However, our proposed approach would still yield less error in search performance, on average, compared to entropy change maximization.

- The average error for the mowing-the-lawn approach is smaller than the average error for the entropy change maximization method. This is because the mowing-the-lawn approach visits A1 that has no uncertainty in the environment while the entropy change maximization method visits A3 where the environmental uncertainty is greatest. However, as the environment in A1 is the least informative, the average actual search performance for the mowing-the-lawn approach is the worst among all methods.

- Note that Fig. 3.3b.1 and Fig. 3.3b.4 are identical. This is because the search locations selected by the proposed approach (in Fig. 3.2a) are the same locations selected when environment information is not available (in Fig. 3.2d) for given search area characteristics. However, the anticipated search performances for these two cases are different. Indeed, comparing Fig. 3.3a.1 with Fig. 3.3a.4 shows that the anticipated search performance when environment information is available is significantly more accurate than the anticipated search performance when there is no environment information. Hence, a benefit of characterizing the environment is to better anticipate the true search performance.

3.6.2 Each sensor on separate vehicles

The results when search and environmental characterization tasks are performed on separate vehicles are plotted in Fig. 3.4. Again, the left plot is the percentage of occurrences of the error in search performance, and the right plot is the percentage of occurrences of actual search performance. The subplots from top to bottom are the results when 1) the locations

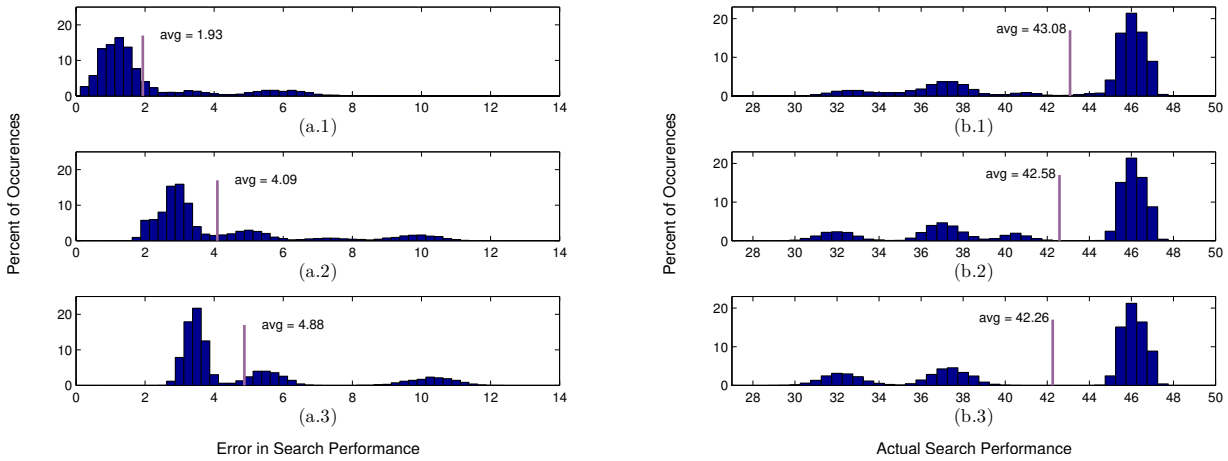


Figure 3.4: Percentage of occurrences for (a) error in search performance and (b) actual search performance when search and characterization are performed on *separate vehicles*. From top to bottom, (a.1) and (b.1) correspond to the proposed approach, (a.2) and (b.2) correspond to the entropy change maximization method, and (a.3) and (b.3) correspond to the case where environment information is not available. Note that the horizontal axis is the negative log of the results. Smaller values for (a) imply less error in search performance and larger values for (b) imply better search performance.

that yield the greatest reduction of uncertainty in search performance are characterized, 2) the locations that maximize the entropy change are characterized, and 3) there is no environmental characterization and the search vehicle plans its path by using the prior environmental distributions. We note that Fig. 3.4a.3 and Fig. 3.4b.3 are the same plots given in Fig. 3.3a.4 and Fig. 3.3b.4, and we show them here for convenience of comparison. It is seen that

- The average error is significantly smaller when environmental characterization is performed at the locations selected by our proposed approach. With respect to the case where there is no environment information (in Fig. 3.4a.3), our proposed approach achieves 60% error reduction while entropy change maximization achieves only 16%, on average. Note that the distribution in Fig. 3.4a.2 is very similar to the distribution in Fig. 3.4a.3. This is because entropy change maximization fails to improve the performance of a follow-on search mission since it leads the vehicle to explore the parts of

the search area that are less likely to be searched. On the other hand, our proposed strategy to select the characterization locations successfully reduces the error in search performance. This is expected since our approach directly penalizes the variation from the true search performance.

- The average error when the sensors are on different vehicles is higher than when both sensors operate on the same vehicle since the search vehicle may search the locations that are not characterized. On the other hand, this results in average actual search performance to be better since the search vehicle can skip the locations that are characterized and found to be uninteresting for search.

The results of Monte Carlo simulations show that our proposed approaches to select the characterization locations outperform the other strategies that frequently exist in the literature. We note that the case where the characterization vehicle and the search vehicle operates simultaneously is a subtle modification of the case where characterization precedes search that we illustrate here, and the corresponding path would be the same path that is shown in Fig. 3.2e. Note that for each characterization location in Fig. 3.2e, depending on the acquired environmental information, the search vehicle either does not sample from that location or visits that location after it is characterized. Hence, the expected gain of characterizing these locations will be the same regardless of whether characterization precedes search or both vehicles perform simultaneously.

Chapter 4

Switching Between Search and Characterization

In this chapter, we consider that the search vehicle is equipped with a search sensor and an environment characterization sensor, but these sensors cannot operate simultaneously. Thus, when the vehicle visits a location, it either actives the search sensor to observe the number of targets at a location, or it activates the environment sensor to observe the environment at that location. The objective of the search mission is to reduce the risk of incorrectly estimating the number of targets below a desired level of risk. We say that a search mission is successful if a desired probability of attaining the desired risk is achieved. We assume switching between search and characterization has no cost and we terminate the mission when the time/distance limit is met, or we find out that the mission goal cannot be accomplished. In Section 4.1, we provide a brief review of the literature on similar problems. In Section 4.2, we present our approach to compute when and where to activate the search sensor and when and where to activate the environment sensor so that a desired level of risk can be attained. In Section 4.3, we show the results of the numerical illustrations.

4.1 Related work

In search applications where the search agent is equipped with a search sensor and an environment characterization sensor, it is often the case that these two sensors can operate

simultaneously. However, there are specific cases where only one sensor can be active due to the physical limitations of the on-board sensors or the computational limitations associated with signal processing. In the context of subsea search (see, for instance, [57] for an application survey), marine systems have employed both acoustic side-scan sonar sensors (search sensor) and acoustic sub-bottom profilers (environmental characterization sensor). Simultaneous data collection is possible in this scenario because the frequency bands of the sensors do not overlap (high frequency and low to mid frequencies, respectively). In Section 3.4, we offer path planning strategies for such cases. However, in some scenarios, simultaneous data collection may not be possible. For example, in order to detect buried objects, some sensing approaches operate at low to mid frequency bands [58]. Similarly, operating in low to mid frequencies has shown increased imaging performance using synthetic aperture sonar processing [59]. In both of these approaches, the search sensor would overlap with typical sub-bottom profiling sensors resulting in decreased performance due to mutual interference. To address this, measurements may be taken separately in order to avoid decreased sensor performance. We note that the approach to compute the optimal paths for such cases is fundamentally different than the approach reported in Section 3.4.

When the search agent can either acquire environment data to reduce the uncertainty in the environment or engage in search to estimate the number of targets in the environment, but not both activities simultaneously, the search problem poses an exploration-exploitation trade-off. Exploration implies increasing our knowledge about the environment, and exploitation implies using the current knowledge of the environment to more efficiently conduct the search mission. The search agent explores the search environment when it performs environment characterization, and exploits its current knowledge of the environment when it performs search. When the search agent searches a location, it returns with an uncertain reward drawn from a known distribution. On the other hand, when the agent characterizes the environment at a location, the expected payoff is zero, but it learns more about the distribution that the reward of a search visit to that location is drawn from. In the search literature, the exploration-exploitation trade-off is sometimes modeled through the well-known multi-armed bandit problem where the decision-maker is faced with the decision of either exploiting current knowledge to improve current utilization or exploring other alter-

natives to increase future utilization, (see for example [60–62]). However, we note that the problem we describe here is fundamentally different and thus it cannot be modeled through a multi-armed bandit problem.

The exploration-exploitation trade-off is also a well-studied problem in other disciplines. In [63], the authors present a framework in monetary problems to help the decision-maker determine whether buying the information to reduce the uncertainty in the outcome outweighs the cost of buying the information. The behavioural study in [64] considers the case where the decision-maker chooses between buying information that may decrease the expected loss and buying information that may increase the expected gain. However, meeting a certain goal, such as attaining a desired level of risk, is not the objective of these studies. In [65], the author presents a method to allocate the funds in an investment portfolio where the objective is to collect a certain amount of reward. However, the results are applicable only for a small class of rewards; when the total sums of the rewards are normally distributed or when each reward obeys a Poisson distribution. Thus, new insights into the exploration-exploitation trade-off, as those attaining a target reward with a non-specific rewards distribution developed in our work, have a broad impact in applications as diverse as finance and health-care.

In some search missions, when searching the area further will not improve the search results, it might be important to terminate the mission to free up the search agent to perform other tasks. There are a few studies in which the search mission is stopped when there is adequate information on the presence or absence of a target (see, for example, [66]). However, in these studies, the search continues until adequate information is acquired. In this work, we terminate the search mission when we determine that adequate information on the number of targets cannot be acquired. This strategy improves search efficiency in cases where the goal of the search mission can be achieved under only certain environmental conditions. In other words, the search mission can be terminated early if the environment is such that the search sensor cannot perform well enough to meet the goals of the mission. To the best of our knowledge, this is the first study that addresses the problem of attaining a desired level of risk and stopping the mission when the desired risk is found to be unachievable.

4.2 Achieving a target level of risk reduction

We consider the scenario where both the search sensor and the environment sensor are placed on the same vehicle, but only one sensor can be active at a time. Thus, when the vehicle visits a location, it either acquires a search measurement by activating the search sensor, or it acquires an environment measurement by activating the environment sensor. The search sensor model and the environment sensor model are presented in Section 2.1 and in Section 3.2, respectively. When a search measurement is acquired, we update the probability distribution of the number of targets by applying the Bayes update rule given in (2.7). Similarly, when an environment measurement is acquired, we update the probability distribution of the environment by applying the Bayes update rule given in (3.1).

Due to the stochasticity in the environment, we may not deterministically know the attained risk reduction after a mission. We instead represent our belief on the attained risk reduction through a probability distribution. This probability distribution maps any possible attained risk reduction that follows from (2.23) to a probability of it being the true value of the attained risk reduction after a mission. We note that characterizing the environmental conditions at a location does not change the attained risk reduction, but it modifies the probability distribution on it. That is, we expect to reduce the uncertainty in the attained risk reduction due to environmental uncertainty rather than to directly increase the attained risk reduction. Thus, the value of acquiring an environment measurement at a location is associated with reducing the environmental uncertainty since it may allow us to better anticipate the attained risk reduction after a mission. We may also choose to characterize the environment when doing so will lead us to better determine when to terminate the search mission. Note that when the environment at a location is deterministically known, the value of performing environmental characterization at that location is zero.

The objective of the search mission is to reduce the risk of incorrectly estimating the number of targets below a desired level of risk $\bar{\beta}$. We note that this objective can also be interpreted as attaining a desired risk reduction

$$\beta = \Big(\sum_{i \in \mathcal{G}} \sum_{j=1}^{m} P\big(E = w_j\big) \rho(i \mid w_j) \Big) - \bar{\beta} \tag{4.1}$$

where $\rho(i \mid w_j)$ is the current risk conditioned on environment w_j (2.21). We note that β in (4.1) can be computed more efficiently than $\bar{\beta}$. Our goal is to determine when to search and when to characterize the environment in order to maximize the probability of attaining the desired level of risk reduction. We assume that switching between the search sensor and the environmental characterization sensor has zero cost and can happen any time during the mission. However, our approach is also applicable if there is a cost associated with switching between the sensors or there is a constraint on when it can happen.

4.2.1 Probability distribution on risk reduction

When the attained risk reduction after a mission is greater than the desired level of risk reduction, we consider that the mission is successfully accomplished. Let y denote the environment measurement acquired at a location and let β represent the desired risk reduction. Because the attained risk reduction in (2.23) is conditioned on the environment w_j, the attained risk reduction is $B\big(i, k \mid w_j\big)$ with probability $P(E = w_j \mid y \in Y)$.

The *probability of success* is the probability of attaining the desired risk reduction and is denoted by $\mathcal{P}$. Given the attained risk reduction conditioned on each environment $w_1, w_2, \ldots, w_m$, we compute the probability of success by

$$\mathcal{P} = \sum_{j:\ B(i,k|w_j)\geq\beta} P(w_j \mid y) \tag{4.2}$$

We again consider a candidate path $\gamma = [q_1, q_2, \ldots, q_N]$. Let $\mathcal{A}_s$ be the action space, and let $a \in \mathcal{A}_s$ be an action the vehicle takes when it visits a cell. Since the vehicle can activate either the search sensor or the environment characterization sensor, the action space is $\mathcal{A}_S = \{\text{search}, \text{characterize}\}$. We denote the sequence of actions taken along the path γ by $\mathbf{a}_\gamma$. Let m_i be the multiplicity of search measurements acquired at the q_ith cell. Then the attained risk reduction when traversing γ is the sum of the attained risk reduction for each cell in the path,

$$B\big(\gamma, \mathbf{a}_\gamma \mid e_\gamma\big) = \sum_{q_i \in \gamma} B\big(q_i, m_{q_i} \mid e_{q_i}\big) \tag{4.3}$$

where $e_\gamma = [e_{q_1}, \ldots, e_{q_N}]$, and $e_i \in w_1, w_2, \ldots, w_m$ is the assumed environment in the ith cell. The notation $(\gamma, \mathbf{a}_\gamma)$ indicates that the attained risk reduction is associated with the path γ and the sequence of actions $\mathbf{a}_\gamma$ taken over the path.

Let y_{q_i} be the environment measurements acquired at the q_ith cell, and y_γ be the set of environment measurements acquired along the path γ. Then, the risk reduction in (4.3) is attained with probability

$$P(e_\gamma \mid y_\gamma) = \prod_{q_i \in \gamma} P(e_{q_i} \mid y_{q_i}) \tag{4.4}$$

Since the true environment in each cell may not be known, we compute (4.3) and (4.4) for each possible set of true environments e_γ. This yields the probability distribution on attained risk reduction conditioned on the set of environment observations $\mathbf{y}_\gamma$. Then the probability of success for traversing γ, taking actions $\mathbf{a}_\gamma$ and observing y_γ is

$$\mathcal{P}_{\gamma,\mathbf{a}_\gamma} = \sum_{e_\gamma:\ B(\gamma,\mathbf{a}_\gamma|e_\gamma)\ \geq\ \beta} P(e_\gamma \mid y_\gamma) \tag{4.5}$$

4.2.2 Gain of selecting a sequence of actions

The optimization problem we address yields a desired path *and* a desired sequence of actions. That is, we seek to determine when and where to search and when and where to characterize the environment. Thus, we are interested in finding the best available path and the best sequence of actions along this path so that the probability of accomplishing the search mission is maximized.

We denote the *desired probability of success* by $\mathcal{B}$. It is the minimum acceptable probability of attaining the desired risk reduction. Thus, the mission is successful if the probability of success $\mathcal{P}$ in (4.5) is greater than or equal to the desired probability of success $\mathcal{B}$. Selecting a path and a set of actions along the path yields a probability distribution on the attained risk reduction conditioned on the environment measurements acquired along the path. Let $\Pi_{B(\gamma)}$ denote the probability distribution in (4.4) on the attained risk reduction in (4.3). We consider that after $\Pi_{B(\gamma)}$ is computed, a decision upon whether the mission is successfully accomplished is made. Let $\delta_\beta : \Pi_{B(\gamma)} \rightarrow \mathcal{A}_D$ be the decision rule that maps the

distribution on attained risk reduction to an action in the action space $\mathcal{A}_D = \{a_0, a_1\}$. The action $a_0 \in \mathcal{A}_D$ represents the decision that the mission will be successful, and the action $a_1 \in \mathcal{A}_D$ represents the decision that the mission will not be successful.

Candidate actions are assessed by evaluating a gain function that results from the utility of forming the decision δ_β. Given decision δ_β when R_a is the true attained risk reduction, we define the corresponding loss

$$U_{\mathcal{A}}\big(R_a, \delta_\beta\big) = \begin{cases} l_1 & \text{if } \delta_\beta = a_0 \text{ and } R_a > \beta \\ -l_2\ N_{\delta_\beta = a_1} & \text{if } \delta_\beta = a_1 \\ -l_3 & \text{if } \delta_\beta = a_0 \text{ and } R_a \leq \beta \end{cases} \tag{4.6}$$

where $l_1 > 0, l_2 > 0$, $l_3 > 0$, and $l_2 << l_1$, $l_2 << l_3$.

We note that the utility function (4.6) does not yield a decision. Rather, it is used to evaluate the effect of selecting among the actions search and environmental characterization. For the decision $d = a_0$, corresponding to the mission being successfully accomplished, there is a positive utility if the true attained risk reduction is greater than the desired risk reduction, and there is a negative utility (cost) if the true attained risk reduction is less than the desired risk reduction. The negative utility represents the severe consequences of incorrectly estimating that the mission is successfully accomplished. For the decision $\delta_\beta = a_1$, corresponding to a mission not being successful, the associated cost is proportional to the traversed path length until the decision is formed ($N_{\delta_\beta = a_1} \leq N$). Incurring a cost in such cases promotes early termination of the mission when the mission cannot be accomplished under the present environmental conditions.

Given a path γ, let $\mathbf{A}_\gamma$ represent the set of possible sequences of actions we can take along the path, and let $B\big(\gamma, \mathbf{a}_\gamma \mid e_\gamma\big)$ be the attained risk reduction conditioned on the set of environments e_γ. Then, for each sequence of actions $\mathbf{a}_\gamma \in \mathbf{A}_\gamma$, the gain of taking these actions is

$$G(\mathbf{a}_\gamma) = \max_{\delta_\beta} \sum_{y_\gamma} \sum_{e_\gamma} P(e_\gamma, y_\gamma)\; U_{\mathcal{A}}\big(B(\gamma, \mathbf{a}_\gamma \mid e_\gamma), \delta_\beta\big) \tag{4.7}$$

Let Ω_γ denote the finite collection of candidate paths available to the vehicle. Then, the optimal path and the best sequence of actions are

$$(\gamma^\star, \mathbf{a}^\star) = \arg\max_{\gamma \in \Omega_\gamma} \Big(\arg\max_{\mathbf{a}_\gamma \in \mathbf{A}_\gamma} G(\mathbf{a}_\gamma) \Big) \tag{4.8}$$

subject to the budget constraint in (2.17).

4.2.3 Reducing computational complexity of the solution

Computing the optimal path and the optimal set of actions in (4.8) is equivalent to determining when and where to search and when and where to characterize the environment. However, maximization of (4.8) is computationally prohibitive when the search space is large, making the proposed approach infeasible in real-time applications. Thus, we briefly comment on computational issues.

Branch-and-bound methods are commonly applied in large state-space optimization problems to reduce the computational complexity of the solution (see Section 6.1.1). However, for the specific problem considered in this paper, there are two drawbacks of applying the branch-and-bound approach. First, computing a meaningful upper bound on the probability of success for a given node can be computationally very challenging. Second, an additional visit to a cell modifies the probability distribution on the attained risk reduction of that cell (to be specific, an additional search visit modifies the range of the probability distribution, and an additional characterization visit modifies the shape of the probability distribution). Thus, in order to update the probability distribution on the attained risk reduction of a path when an additional visit is made to a cell in this path, it is necessary to keep track of how the previous visits to that cell affected the probability distribution up to that visit. Thus there are significant memory requirements and corresponding computational requirements.

Instead, we apply a simple trick to reduce the computational effort. We quantize attained

risk reduction of a path into N_0 possible values and normalize attained risk reduction to 1. This allows us to compactly represent the probability distribution on the attained risk reduction, and we no longer need to store the risk values but only the probability distribution on them. With N_0 risk reduction values, which is independent of the path length N and the number of possible environments m, we represent the probability distribution on the attained risk reduction with only N_0 values. This not only reduces the required memory storage, but also significantly speeds up the corresponding computations since the number of addition and multiplication operations are drastically reduced. For example, when a new cell is added to a path of length N, we perform mN_0 addition and mN_0 multiplication operations instead of m^{N+1} addition and m^{N+1} multiplication operations. Using this method results in computing the probability of success on a desired risk reduction of $\beta - \frac{1}{N_0}$ which is negligibly different than β when N_0 is large. Alternatively, one can modify the desired risk reduction as $\beta + \frac{1}{N_0}$ to account for the effect of quantization.

In addition, we pre-compute all probability distributions or the attained risk reduction at each cell corresponding to all possible environment measurements and number of search visits. Then given the path, the actions along the path and the environment measurements (if any), we compute the attained risk reduction and the corresponding probability distribution on it by using the pre-computed values for each cell along the path.

While these methods reduce the computational time from several hours to several minutes, the proposed approach can still be computationally infeasible for large-scale problems. In on-going work not reported herein, we are pursuing ways to efficiently reduce the search space by pruning the paths that are guaranteed to be not the optimal.

4.3 Numerical results

In this section, we present simulation results that illustrate the efficacy of the proposed strategy for attaining the desired risk reduction. We consider that the search agent is equipped with a search sensor and an environment characterization sensor, but that these sensors cannot operate simultaneously. Simulations are conducted over a 6-by-6 cell search area.

When the vehicle visits a location, it either activates the search sensor to observe the

number of targets or the environment characterization sensor to observe the environmental conditions at that location. The probability of detection D, and the probability of at least one false alarm F for each environment are shown in Table 2.1. Sensor performance increases with increasing probability of detection and decreases with increasing probability of false alarm. To illustrate the relationship between sensor performance, the probability of false detection, and the probability of correct detection, the attained risk reduction corresponding to searching a location that has one of each of the three environment types is shown in Table 4.1 when the relative costs of overestimating and underestimating the number of targets are $c_1 = 3$, $c_2 = 1$. We also show the attained risk reduction for searching that location a second time. Larger attained risk reduction implies better sensor performance. When the vehicle characterizes the environment at a location, it acquires an environment measurement with respect to the characterization sensor model and the true environment at the location. We use the environment characterization sensor model in (3.51) with $a_{11} = 0.9$, $a_{22} = 0.92$, $a_{33} = 0.94$, and $a_{ij} = a_{ik}$ for $j, k \neq i$. That is, for example, there is 0.9 probability of acquiring environment measurement $Y = w_1$, 0.05 probability of acquiring environment measurement $Y = w_2$, and 0.05 probability of acquiring environment measurement $Y = w_3$ when w_1 is the true environment at the location.

Although types of possible environments are known, the specific environment at any location is uncertain. Fig. 4.1 shows the search area and the corresponding probability distribution for each cell. The number in each cell is simply a label for each cell. For each distribution $[p_1, p_2, p_3]$, p_j is the probability that the environment in the cell is w_j. For example, Fig. 4.1 indicates that there is a 0.95 probability that the environment in cell 11 is w_1, and a 0.05 probability that the environment is w_3. Note that the lighter cells are

Table 4.1: Attained risk reduction with deterministic environments

Environment	Attained risk reduction for single search pass	Attained risk reduction for two search passes
w_1	0.196	0.337
w_2	0.356	0.559
w_3	0.824	0.924

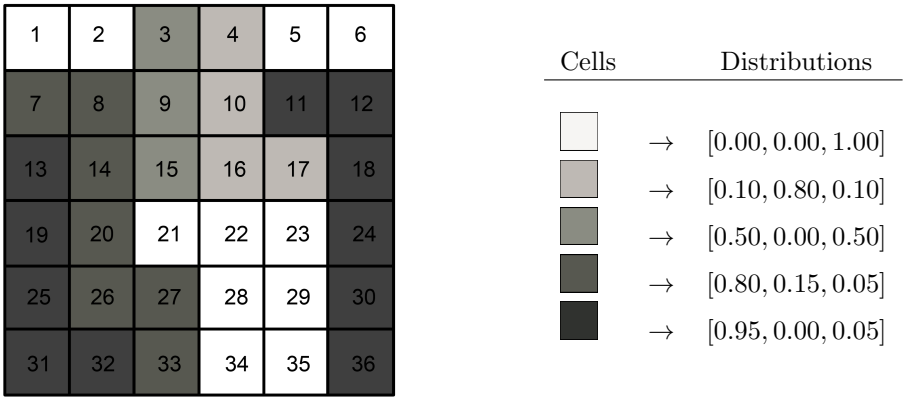


Figure 4.1: Search area and cell-wise environment distributions

more likely to belong to environment w_3, and the darker cells are more likely to belong to environment w_1. The greatest uncertainty about the environment occurs in cells 3, 9 and 15.

Suppose the vehicle visits cell 3 twice. It can either observe the number of targets in the cell twice or observe both the number of targets and the environmental conditions in the cell once. Based on the results in Table 4.1, since the prior environment distribution for cell 3 assigns equal probabilities to environments w_1 and w_3, the attained risk reduction for searching the cell twice is either 0.337 or 0.924 with equal probabilities. Now, suppose the vehicle searches the cell once, then characterizes the environment in the cell and observes $Y = w_3$. Then, the attained risk reduction is either 0.824 with a probability of 0.95 or 0.196 with a probability of 0.05. Thus, the benefit of searching the cell twice is the increase in the attained risk reduction – therefore, decrease in the resulting risk – conditioned on the environment, and the benefit of characterizing the environment is the reduction in the uncertainty of the attained risk.

The objective is to find the best path and the best sequence of actions to attain a desired level of risk reduction. When the best path is computed, the vehicle visits the first cell of the best path and executes the corresponding action. After acquiring data, either on the number of targets or on the environmental conditions, we update the corresponding distribution for that cell and re-plan the best path and best sequence of actions for the remaining mission

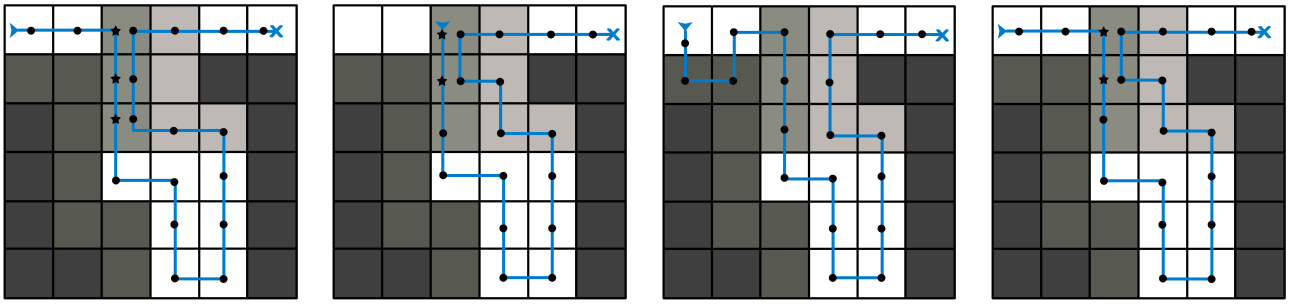
length using the new information. Suppose the vehicle searches the ith cell and acquires the measurement z. Then, with a probability of $P(E = w_j)$, the *achieved* risk reduction in cell i is

$$\rho(i) - \rho(i \mid z, w_j) \tag{4.9}$$

which yields a probability distribution on the achieved risk reduction. When re-planning, the attained risk reduction of a path in (4.3) and its probability distribution in (4.4) are modified accordingly to account for the probability distribution on achieved risk reduction in cell i. We will now show the numerical results for simplistic scenarios.

We assume the vehicle starts the mission in cell 1. We note that for this particular problem the vehicle is not constrained to move forward to the next grid cell in a row. Instead, it can move in four directions - up, down, left or right - as long as it remains in the search area. If the equipped sensors necessitate the motion constraint described in Section 2.3, we simply remove those candidate paths that do not satisfy the motion constraint. The mission is terminated either when the maximum mission length is met or when the desired level of risk reduction cannot be attained under present environment conditions. For numerical illustrations, we consider three different mission objectives to show how the objective of a mission, a desired probability of attaining a desired risk reduction, affects the best path and the best set of actions. For all illustrations, the mission length is 20 and the relative costs in (4.6) are $l_1 = l_3 = 1, l_2 = 0.01$.

We first consider $\beta = 13$ and $\mathcal{B} = 0.85$. For a mission length of 20, this implies that at least 0.85 probability of attaining, on average, a 0.65 risk reduction per cell is required. Note that the attained risk reduction for a single search visit is given as 0.824 in Table 4.1 even when the environment is the most informative environment w_3. The best path and the best set of actions corresponding to this case are shown in Fig. 4.2a. The blue solid line represents the best path for the vehicle, the circles represent search actions, and the asterisks represent the characterization actions. We see that cells 3, 9 and 15 are first characterized and then searched while the other cells in the path are searched once. There is a large uncertainty in the environment for cells 3, 9 and 15. The attained risk reduction for these cells might be either very small so that the desired risk reduction cannot be attained with the desired probability, or it might be large enough to meet the mission objective. Thus, accomplishing



(a) (b) (c) (d)

Figure 4.2: Best path and best sequence of actions when (a) $\beta = 13$ and $\mathcal{B} = 0.85$, (b) $\beta = 13$ and $\mathcal{B} = 0.85$ and $z_1 = z_2 = 2$ are observed, (c) $\beta = 11$ and $\mathcal{B} = 0.85$, and (d) $\beta = 23$ and $\mathcal{B} = 0.65$.

the mission is conditioned on the environment observations acquired from cells 3, 9 and 15. The mission objective can be satisfied only if the environment w_3 is observed in each of these cells. If a different environment is observed in any of these cells, the mission cannot be accomplished and the vehicle terminates the mission. Note that the cells 3, 9 and 15 are characterized as early as possible to promote early termination of the mission if the mission cannot be accomplished under present environmental conditions. This is due to the cost $l_2 N_{\delta_\beta}$ in (4.6), which is proportional to the length of the traversed path before terminating the mission.

Suppose the vehicle searches cell 1 and cell 2 and acquires *good* search measurements that yield a greater risk reduction than expected to be attained. For example, in our simulations, when the deterministic environment is w_3, acquiring search measurement $z = 2$ yields an achieved risk reduction of 0.88 while the attained risk reduction prior to acquiring the measurement is 0.824. After acquiring search measurements, the vehicle computes the achieved risk reduction in (4.9) for both cells. When re-planning a new path with the remaining mission length, a smaller risk reduction is required to be attained since a greater risk reduction than expected is achieved in cells 1 and 2. The corresponding path and the set of actions for this specific case are shown in Fig. 4.2b. Note that the resulting path is different from the path in Fig. 4.2a. Instead of characterizing the environment in cell 15, the vehicle searches cell 10 so that accomplishing the mission is now conditional on observing environment w_3 in only cells 3 and 9, but not in cell 15. It is evident that observing environment w_3 in cells

3 and 9 occurs with greater probability compared to observing environment w_3 in cells 3, 9 and 15.

In our second illustration, we consider a lower risk reduction $\beta = 11$ that should be easier to achieve and the same desired probability of success $\mathcal{B} = 0.85$. By lowering the desired risk reduction, we expect that the vehicle chooses search actions more often compared to achieving a higher risk reduction. That is, lowering the desired risk reduction increases the probability of attaining it with a fewer number of characterization actions so that accomplishing the mission is conditioned on fewer environment measurements. The corresponding path and set of actions are shown in Fig. 4.2c. We see that the vehicle performs only search actions. Environmental characterization actions are not required since the probability of success can be met even if some environments correspond to poor sensor performance. Hence, accomplishing the mission is not conditioned on specific environment measurements.

We finally consider the case when $\beta = 13$ but the required probability of success is lowered to $\mathcal{B} = 0.65$. The effect of lowering the desired probability of success is similar to the effect of lowering the desired risk reduction. However, these two cases, lowering the desired probability of success and lowering the desired risk reduction, can yield different paths and different sets of actions depending on the search area characteristics. We again expect that the vehicle conducts fewer environment characterizations compared to a higher value of desired probability of success. Fig. 4.2d shows the resulting path and the set of actions. Compared to Fig. 4.2a, the environment in cell 15 is not characterized so that accomplishing the mission is conditioned on acquiring environment measurement w_3 only in cells 3 and 9.

We note that the corresponding path and set of actions for the first illustration result in a positive gain in (4.7) when the desired risk reduction or the desired probability of success is lowered as in the later illustrations. However, since the gain of taking a path and a set of actions in (4.7) depends on the probability of acquiring a particular set of environment measurements along the path $P(y_\gamma)$, the path in Fig. 4.2a is suboptimal and therefore not preferred.

Chapter 5

Multi-Vehicle Search Problem

In this chapter, we consider that there is a team of k collaborating vehicles to perform the search mission. Each vehicle is equipped with a search sensor and the goal is to collaboratively maximize the joint search performance. We employ the partially observable Markov decision process (POMDP) framework to compute optimal strategies. We specifically address the cases where vehicle-to-vehicle communication is possible, but communication is difficult and incurs a mission-relevant cost, and thus there is a trade-off between the cost of communication and its value. In Section 5.1, we provide the formal definition of a POMDP and its multi-agent extension. In Section 5.2, we provide a brief literature review of multi-agent planning problems. We introduce the multi-vehicle search problem that we consider in this study in Section 5.3. Briefly, we consider the specific case where communication is range-limited, and therefore the communicating vehicle may need to maneuver towards the other vehicle's communication range in order to communicate. In Section 5.4, we present the results of our numerical illustrations for simplistic scenarios.

5.1 Background

The single-agent planning problem under uncertainty is sometimes modeled as a partially observable Markov decision process (POMDP). A POMDP can be formally represented by a tuple $\langle S, A, T, O, Z, R \rangle$ where S is the state space, A is the set of actions, T is the set of transition probabilities i.e. the probability of transitioning to a new state from the

current state when taking a particular action, O is the set of observation probabilities i.e. the probability of seeing an observation when moving to a new state after taking a particular action, Z is the set of observations and R is the set of rewards. A state $s \in S$ fully describes the environment of the problem. When taking an action $a \in A$, the problem transits to a new state s' from the current state s with a probability of $T(s, a, s') = P\big(s' \mid s, a\big)$, and we acquire an observation $z \in Z$ with a probability of $O(s', a, z) = P\big(z \mid s', a\big)$. For each state-action pair, there is an associated reward $R(s, a)$ for taking action a in state s. Due to uncertainty in the observations, the true state of the problem cannot be known with certainty. Instead, the belief on the true state of the problem is represented through a belief distribution over the state space, referred to as the *belief state*. The goal is to find the optimal policy – a mapping from the belief states to actions – that yields the maximum attainable reward.

In general, the POMDP approach suffers from two structural properties of the planning problem: the curse of dimensionality, i.e. the number of computations grows exponentially with the size of the state space, and the curse of history, i.e. history of actions and observations grows exponentially with the planning horizon. Due to these two limitations, POMDP solutions become intractable for many real-world problems. Solving a finite-horizon POMDP is known to be PSPACE-complete (polynomial space complete) [67]. However, recent works in the literature address the computational complexity of solving POMDPs, and there are some notable approaches that yield near-optimal policies in feasible time [68–72].

In the literature, there are a number of methods that extend the POMDP framework to a multi-agent setting. Examples include partially observable identical payoff stochastic games (POIPSG) [73], multi-agent team decision problems (MTDP) [74], interactive POMDPs (I-POMDP) [75], and decentralized POMDPs (Dec-POMDP) [76]. Among all, Dec-POMDP method received the most attention and numerous variants of the method have been proposed. In a Dec-POMDP framework, the action space is the set of joint actions $\vec{a} \in \vec{A}$ where $\vec{A} = A_1 \times A_2 \times \ldots \times A_k$ and A_i is the set of actions for the ith agent. Similarly, the observation space is the set of joint observations $\vec{z} \in \vec{Z}$ where $\vec{Z} = Z_1 \times Z_2 \times \ldots \times Z_k$. Transition probability $T(s, \vec{a}, s')$ is the probability of transitioning to a state s' after taking the joint action $\vec{a}$ at state s, and the observation probability $O(s', \vec{a}, \vec{z})$ is the probability of receiving the joint observation $\vec{z}$ after taking the joint action $\vec{a}$ and ending up at state s'. In

Dec-POMDPs, agents compute a joint policy that maximizes a joint reward, and each agent executes its own policy and receives local observations.

Due to the added complexity of reasoning about the other agents' actions and observations, the solution to a Dec-POMDP problem becomes easily intractable for a large number of agents and/or a large planning horizon. The problem is previously shown to be NEXP-complete (non-deterministic exponential time) [76]. There is currently no work in the literature that offers a scalable solution to the Dec-POMDP problem. Even the state-of-the-art solutions make strong assumptions on the problem domain such as transition-independence between the agents [77], sparse interactions in joint policy spaces [78] or availability of macro actions [79].

5.2 Related work

Bernstein *et al.* proposed the Dec-POMDP framework to compute the joint plans for a team of collaborative agents [76]. They consider that the centralized policies for each agent are computed in plan-time (i.e. prior to policy execution), and then, each agent updates its policy based on the local observations it receives in run-time (i.e. during the policy execution). Pynadath and Tambe [74] and Goldman and Zilberstein [80] extended Bernstein *et al.*'s work to account for explicit communication between the agents in order to maintain team coordination and improve joint performance.

Communication can be either considered as a domain action that the agent chooses over other actions [81–83], or it can be considered as a separate action where the agent chooses communication and domain actions concurrently [80, 84–87]. In a vast majority of the Dec-POMDP literature, it is assumed that when an agent chooses to communicate, all agents broadcast their local observations to each other in order to synchronize their world view (see, for example, [87–89]). However, in many real-world applications, broadcasting information is not feasible and only agent-to-agent communication is available. There are surprisingly few studies in the literature that consider agent-to-agent communication. Notable examples include [82, 83, 90, 91]. In these studies, an agent not only chooses *when* to communicate, but it also chooses with *whom* to communicate.

A major challenge of employing the Dec-POMDP framework is the computational complexity of the solution. It has been proved that the Dec-POMDP problem with and without communication is NEXP-complete [74,76]. In most cases, either the solution is computed offline (e.g. prior to plan execution), or it is computed online (e.g. during the plan execution). When the solution is computed offline, all computations to compute the domain actions and the communication actions are performed in plan-time [74, 80, 81, 87, 92], and when the solution is computed online, all computations are performed in run-time [82, 83, 89, 93]. An alternative approach is the hybrid solution where the domain actions are computed in plan-time and communications actions are computed in run-time [90, 91].

The offline solution becomes intractable even for toy examples since computations should be performed for all possible situations including all possible sets of observations. In addition to the computational complexity, the memory requirements of the offline solution can also be substantially large. Often, the computational complexity is reduced by applying either a top-down heuristic search [94, 95] or a bottom-up dynamic programming approach [96–98]. Seuken and Zilberstein proposed the state-of-the-art memory-bounded dynamic programming algorithm (MBDP) which combines the top-down heuristic search and the bottom-up dynamic programming to efficiently prune the search space so that both the solution complexity and the memory requirements can be significantly reduced [99]. Later, Amato *et al.* proposed an incremental policy generation algorithm to further speed-up the dynamic programming approach [100]. In [100], the authors also presented a comparison of the computation time and the solution quality for a number of optimal and approximate dynamic programming approaches based on the results of small-scale benchmark problems (e.g. meeting in a grid, box pushing, stochastic Mars rover). The results show that even the state-of-the-art offline Dec-POMDP approaches suffer from scalability to larger domains and to larger planning horizons.

Compared to offline solutions, online solutions and hybrid solutions should be implementable in real-time. In online solution, all computations are performed during execution time. It is often the case that the solution becomes intractable unless the planning horizon is sufficiently small. Often, a myopic planning approach is considered where the agents compute one-step optimal action at a time [89]. Williamson *et al.* present a non-myopic

approach, but the planning horizon in their work (a horizon of 5) is still very small for real-world applications [82, 83]. Hybrid solutions combine online planning with offline planning. In hybrid solutions, domain actions are computed in plan-time, and communication actions are computed based on the received observations in run-time. Roth *et al.* [90, 91] propose a hybrid approach where centralized policies for each agent are computed offline assuming free communication between the agents, and then decentralized policies that reason about communication actions are computed online based on the local observations.

In multi-agent planning problems, determining when an agent should communicate its history of local observations with another agent is a notoriously difficult problem. It is often not feasible to compute the cost and the value of communication in every planning instance. In the literature, the question of when-to-communicate has been mostly overlooked or over-simplified. The vast majority of studies aim to reduce the number of times the agents choose to communicate, and they devise simple heuristics instead of explicitly accounting for the cost of communication. The lack of exact valuation of communication may result in poor performance due to either communicating too often or not frequently enough. Nair *et al.* [81] propose that an agent should communicate its local observations at every K steps or less. However, their aim is not to improve team performance, but to reduce the computational requirements and memory requirements of computing joint policies. Roth *et al.* [90] propose that communication should be selected in the case that the optimal actions before and after communication events would be different. A more principled approach is proposed by Williamson *et al.* [83] where the authors employ a reward shaping function to heuristically compute the value of communication. In [83], the value of communication is the amount of change in the agent's current belief about the state of the world compared to the last synchronized belief state. That is, the value of communication is directly computed from the amount of mis-coordination between the agents. Wu *et al.* [89] propose that communication should be selected when an observation with significantly low probability of being observed is received by the agent. Their approach also aims to penalize belief discrepancy between the agents similar to Williamson *et al.* [83]. More recently, Unhelkar and Shah [93] proposed an online solution where the agents compute the rewards for maintaining the current policies, for updating the policies without communication, and for updating the policies with com-

municated information. Then, the agent selects communication only when the reward for the case of updating the policies with communicated information is greater than the other two. We note that none of these approaches guarantees that communication will improve the team performance since the exact cost of communication is not factored in. Specifically, when the cost of communication is very large, the proposed approaches are likely to result in poor performance.

5.3 Multi-vehicle search problem

Our goal is to extend our approach for the single-vehicle search problem described in Chapter 2 to a multi-vehicle setting. We assume that there is a team of k collaborating vehicles whose goal is to maximize a joint reward. We consider that vehicle-to-vehicle communication is possible, but communication incurs a cost. Communication range is bounded, and there is a known communication range for each vehicle within which it can communicate with other vehicles. Thus, when a vehicle chooses to communicate with another vehicle, it may need to abandon its current task and maneuver towards the other vehicle to be within its communication range. In such cases, the cost of communication is dominated by the effort applied to maneuver towards the other vehicle's range as well as the loss associated with abandoning the current task. We believe vehicle-to-vehicle bounded communication better fits into real-world scenarios since the search space is often too large to communicate anytime and only the vehicles that are close enough to each other can communicate. For instance, in subsea applications, since communication is unreliable and communication bandwidth is very low, the cost of communication can be very large. Because communication range is limited, an autonomous underwater vehicle (AUV) may need to abandon its current task to move within range of another AUV or communication node. Failure to adequately account for the true cost of communication can result in poor performance. When a vehicle is within the communication range of another vehicle, we assume instantaneous, noise-free communication. However, we note that our approach can be easily modified to address cases where information is transmitted through noisy communication channels.

As in any multi-agent problem, a major challenge is to find a tractable solution that

scales well with the number of agents. A specific question that our work addresses is how to optimally determine when an agent should communicate its history of local actions and observations with another agent. Unlike the prior approaches in the literature, our approach accounts for the exact value and cost of communication. Computing the value and the cost of communication can be computationally expensive due to a large number of vehicles and a large planning horizon. Thus, we instead employ an efficient heuristic to compute when the value of communication can be larger than its cost. This reduces the computational complexity of the problem drastically, and it scales well with the number of vehicles.

Since the Dec-POMDP approach is computationally complex, we are interested in finding more scalable approaches that better fit our problem. Thus, instead of planning for the optimal trajectories and optimal communication actions at the same time, we separate the communication planning problem from the path planning problem. That is, we first compute the optimal paths for each vehicle for a finite horizon prior to execution time, and then, we generate policies to optimally determine when a vehicle should communicate with another vehicle during the execution time. We note that our approach of separating path planning from communication planning is similar to the hybrid solution proposed by Roth *et al.* [90] since path planning takes place offline and communication planning takes place online. However, unlike Roth *et al.*, we compute optimal paths instead of optimal policies for each vehicle. Thus, the computational requirements and the memory requirements of our solution are substantially smaller and we achieve much better scalability with the number of vehicles and the planning horizon.

Given the pre-computed optimal paths for each vehicle, the vehicles choose when to communicate with another vehicle during online planning. A communication action is preferred only when the value of communication is greater than the cost of communication. When the vehicles are close enough to each other to communicate, they both communicate and perform their pre-computed domain actions at the same time. However, when a vehicle has to maneuver towards another vehicle to communicate, it must abandon its pre-computed domain action if it chooses a communication action.

5.3.1 Computing optimal paths in plan-time

We initially compute the optimal paths for each vehicle prior to execution time. We specifically consider that the objective of the search mission is to minimize the joint risk associated with incorrect estimation of the number of targets as in Section 2.2.2. Hence, given that there are k search vehicles, we are interested in finding the optimal paths $\gamma_\star^{[1]}, \gamma_\star^{[2]}, \ldots, \gamma_\star^{[k]}$ such that

$$\gamma_\star^{[1]}, \gamma_\star^{[2]}, \ldots, \gamma_\star^{[k]} = \underset{\gamma^{[1:k]} \in \Omega_\gamma}{\arg\max}\, B\big(\gamma^{[1:k]}\big) \tag{5.1}$$

where $\gamma^{[i]}$ is a path for the ith vehicle, $\gamma_\star^{[i]}$ represents the ith vehicle's optimal path, and $B\big(\gamma^{[1:k]}\big)$ is the joint risk reduction in (2.25).

Computing the optimal paths in (5.1) can be expensive when the number of vehicles is large. Thus, we instead propose a sequential path planning approach where each vehicle's path is computed one by one given the previously computed paths of other vehicles. That is, we start with computing the optimal path for the first vehicle

$$\gamma_\star^{[1]} = \underset{\gamma^{[1]} \in \Omega_\gamma}{\arg\max}\, B\big(\gamma^{[1]}\big) \tag{5.2}$$

and then, given the previously computed paths for vehicles $1, 2, \ldots, i-1$, we compute the optimal path for the ith vehicle

$$\gamma_\star^{[i]} = \underset{\gamma^{[i]} \in \Omega_\gamma}{\arg\max}\, B\big(\gamma_\star^{[1:i-1]} \cup \gamma^{[i]}\big) \tag{5.3}$$

for all $2 \leq i \leq k$.

Sequential allocation of optimal paths reduces the complexity of path planning from $\mathcal{O}(\mathcal{S}^k)$ to $\mathcal{O}(k\mathcal{S})$ where $\mathcal{S}$ is the complexity of computing a single path. Computing the paths sequentially scales well with the number of agents due to a linear increase of the computational effort. However, the scalability of the solution comes at the expense of suboptimal search performance. There are very few studies in the literature that address performance guarantees for the sequential allocation of optimal paths (see, for example, [101, 102]). In both [101] and [102], the lower bound on the performance - the worst-case ratio of the joint

return from sequential allocation to the optimal joint return - is 0.5. That is, when the paths are sequentially computed, the resulting joint reward is guaranteed to be at least 0.5 of the optimal joint reward when paths are simultaneously computed. However, we note that our preliminary results show a substantially lower difference between the performances of the sequentially computed paths and the optimal paths.

When computing the optimal paths in plan-time, we assume that there is no communication between the vehicles. Thus, in general, each vehicle is likely to be assigned to a certain part of the search environment. Indeed, when optimal paths and optimal communication actions are computed simultaneously, the vehicles are likely to stay in close proximity to each other. Thus, when decoupling the path planning problem from the communication planning problem, we add a reward function that factors in the availability of vehicle-to-vehicle communication. We believe adding an appropriate reward function adequately accounts for the effect of non-simultaneous computation of optimal paths and optimal communication actions.

For instance, when there are only two vehicles ($k = 2$), (5.1) can be modified to

$$\gamma_\star^{[1]}, \gamma_\star^{[2]} = \underset{\gamma^{[1:2]} \in \Omega_\gamma}{\arg\max} \Big(B\big(\gamma^{[1:2]}\big) + D\big(\gamma^{[1]}, \gamma^{[2]}\big) \Big) \tag{5.4}$$

where $D\big(\gamma^{[1]}, \gamma^{[2]}\big)$ is the reward function associated with the availability of communication. While our intention is not to model a specific reward function, in the numerical illustrations in Section 5.4 we use the reward function

$$D\big(\gamma^{[1]}, \gamma^{[2]}\big) = \sum_{j:\ d(q_j^{[1]}, q_j^{[2]}) \ \leq \ \epsilon_{\text{comm}}} \alpha_\Sigma - C_\Sigma \tag{5.5}$$

where $\epsilon_{\text{comm}} > 0$ is the communication range, $d(q_j^{[1]}, q_j^{[2]})$ is the distance between the cells the vehicles visit at step j, C_Σ is the cost of communicating the message Σ to the other vehicle and $\alpha_\Sigma > C_\Sigma$ is the reward for communication. The reward in (5.5) is associated with the vehicles staying close enough to each other to communicate a message Σ when following the paths $\gamma^{[1]}$ and $\gamma^{[2]}$.

We note that our path planning approach computes a path instead of a policy for each

vehicle. A path is simply a sequence of actions, while a policy is a complete plan contingent upon all possible observations the vehicle may acquire during a mission. Our intention with computing paths rather than policies for each vehicle is that each vehicle can only receive its local observations during the plan execution and does not know the observations of any other vehicle unless they communicate. Thus, we believe, computing the joint policies contingent upon other vehicles' future actions and observations has small benefit compared to its significantly large computational requirements. On the other hand, the disadvantage of computing a path rather than a policy is that it necessitates an adaptive path planning approach where a vehicle re-plans its path after acquiring an observation. When re-planning can be performed sufficiently fast, our approach is feasible. When re-planning cannot be performed sufficiently fast, we propose that an individual policy computation step follows path planning. That is, after computing the optimal paths for each vehicle, the optimal policy for each vehicle can be individually computed given the planned actions and expected observations of the other vehicles. While this is similar to the algorithm in [81] where the authors compute individual policies one by one and iterate over the policy computation until a Nash equilibrium is found, our approach has a significantly lower worst-case complexity.

5.3.2 Computing optimal communication actions in run-time

As the vehicles explore the environment, they may need to communicate their findings with each other to re-allocate the paths in order to improve the joint performance. We specifically aim to compute the individual communication policies for each agent that show when and with whom to communicate. We note that while our approach does not necessarily minimize the communication that occurs between vehicles, it leads to communication events as infrequent as possible by choosing to communicate only when doing so will improve joint performance.

Communication incurs a cost associated with the effort applied to communicate, and thus, each vehicle has to reason about the cost and the value of communication in run-time. Unlike prior approaches in the literature, we factor in the exact cost of communication when computing the communication policies. Computing the cost of communication can sometimes be expensive as we may need to compute a route to maneuver towards the communication

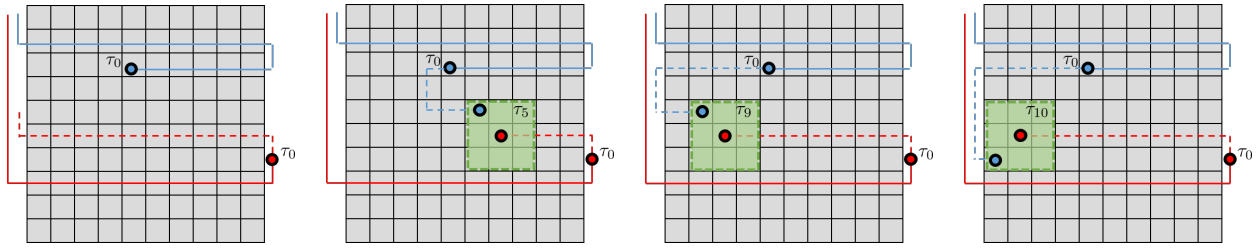


Figure 5.1: Alternative paths for vehicle 1 to communicate with vehicle 2. a) τ_0 is the initial node to reason about communication. Dashed red line is the future trajectory for vehicle 2. Starting at node τ_0, vehicle 1 can take one of the alternative paths b) when taking the shortest path, communication occurs at node τ_5, c) otherwise, it can either occur at node τ_9 or d) at node τ_{10}.

range of the other vehicle. In addition, computing the value of communication can be expensive when the planning horizon is large. Thus, we use a simple but efficient strategy to reduce the number of times we compute the cost and the value of communication.

Given the optimal paths $\gamma_\star^{[1]}, \gamma_\star^{[2]}, \ldots, \gamma_\star^{[k]}$, the multi vehicle search problem with limited communication reduces to the problem of determining when vehicle i should communicate with vehicle j to maximize the joint risk reduction. We use the POMDP framework in Section 5.1 to compute the individual communication policies for each vehicle. Available actions are $A = \{a_0, a_1, \ldots, a_{k-1}\}$ where a_0 is the action of not communicating with any vehicle, and a_j for $j = 1, 2, \ldots, k-1$ is the action of communicating with vehicle j. At every decision instance, vehicle i chooses either $a_j \in A$ to communicate with vehicle j, or it chooses $a_0 \in A$ to not to communicate. When vehicle i chooses not to communicate, it performs its next domain action based on its pre-computed optimal path $\gamma_\star^{[i]}$. When the vehicle chooses to communicate with vehicle j and it is in vehicle j's communication range, communication occurs instantly. However, when vehicle i chooses to communicate with vehicle j and it is not within the range of vehicle j, it computes a new route to enter into the communication range of vehicle j.

Fig. 5.1a illustrates a simplified version of the problem where there are only two vehicles ($k = 2$). The solid blue line and the solid red line represent the trajectories for vehicles v_1 and v_2 up to time τ_0. At time τ_0, vehicle v_1 reasons about communicating its history $\vec{h}$ of

actions and observations with vehicle v_2. The dashed red line represents the remaining part of the optimal trajectory for vehicle v_2, and this trajectory is known to vehicle v_1. Suppose that vehicle v_2 can be communicated with within its 8 neighboring cells. Fig. 5.1b-d show the candidate paths for vehicle v_1 to communicate with vehicle v_2. The green box represents the communication range of vehicle v_2, and the blue dashed line represents the candidate path for vehicle v_1 to enter the communication range of vehicle v_2. Fig. 5.1b shows the shortest distance path with a length of 5. We consider that while the vehicle can take this path that minimizes the effort to communicate with the other vehicle, it does not acquire any observation along this path. Fig. 5.1c and Fig. 5.1d show two other candidate paths with a length of 9 and 10, respectively. When vehicle v_1 takes either of these two paths, it also observes the number of targets at the cells along these paths.

Optimal communication policy

Let $\tau = 0, 1, \ldots, \mathcal{N}$ denote the discrete time, let b_τ represent the belief state at time τ such that $b_\tau(s)$ denotes the probability that the problem is in state $s \in S$ at time τ, and let π be a policy that maps belief states into actions (e.g. $a = \pi(b_\tau)$). We update our belief about the state of the world after taking an action $a \in A$ and acquiring an observation $z \in Z$ by updating the probability distribution

$$b_\tau(s') = \eta O(s', a_\tau, z_\tau) \sum_{s \in S} T(s, a_\tau, s') b_{\tau-1}(s) \tag{5.6}$$

over all states $s' \in S$. Our goal is to find the optimal policy $\pi^\star$ such that for any belief state, the associated action with that belief state yields the maximum attainable reward.

Suppose that vehicle i chooses an action from the action space $A = \{a_0, a_1, \ldots, a_{k-1}\}$ where a_0 represents the action for performing the domain action and a_j represents the action for communicating with vehicle j for $j = 1, 2, \ldots, k-1$. Let $q^{[i]}$ be the cell vehicle i visits when it takes action a_0, and let $z_{q^{[i]}}$ be the observation it acquires from cell $q^{[i]}$. Then, the reward for choosing action a_0 and observing $z_{q^{[i]}}$ is

$$R(b_\tau, a_0, z_{q^{[i]}}) = B\big(z_{q^{[i]}}\big) + \max_{\pi(b_{\tau+1})\in\mathcal{A}} R\big(b_{\tau+1}, \pi(b_{\tau+1})\big) \tag{5.7}$$

where the first term is the achieved risk reduction in (2.25) and it represents the immediate reward for choosing action a_0, and the second term is the future reward corresponding to the best action to be chosen at the next time step $\tau+1$ based on the updated belief state $b_{\tau+1}$. Then, the reward for choosing action a_0 is computed by averaging (5.7) over the observation space

$$R(b_\tau, a_0) = B\big(q^{[i]}\big) + \max_{\pi(b_{\tau+1})\in\mathcal{A}} R\big(b_{\tau+1}, \pi(b_{\tau+1})\big) \tag{5.8}$$

On the other hand, when vehicle i chooses action a_j to communicate with vehicle j, it computes the candidate paths to move into vehicle j's communication range (as shown in Fig. 5.1b-d). Let $\gamma^{[i][j]}$ be a candidate path that vehicle i can take to communicate with vehicle j, and let $\Omega_{\gamma^{[i][j]}}$ be the set of such paths. When vehicle i moves into the range of vehicle j, both vehicles communicate their history of actions and observations denoted by $\vec{h}_{ij}$, and they re-plan their paths based on their synchronized belief about the state of the world. Let $\tilde{\gamma}^{[i]}$ and $\tilde{\gamma}^{[j]}$ be the updated paths after communication for vehicles i and j, respectively. We define the reward for choosing action a_j and taking the candidate path $\gamma^{[i][j]} \in \Omega_{\gamma^{[i][j]}}$ as

$$R\big(b_\tau, a_j, \gamma^{[i][j]}\big) = B\big(\gamma^{[i][j]}\big) + B\big(\tilde{\gamma}^{[i]} \cup \tilde{\gamma}^{[j]} \cup \gamma_\star^{[1:k\setminus i,j]}\big) \tag{5.9}$$

where the first term is the expected risk reduction for traversing the path $\gamma^{[i][j]}$ and the second term is the joint search performance after communication given the re-planned paths for both vehicles. When re-planning the paths, we consider that all other vehicles except vehicle i and vehicle j perform as excepted. We note that when the communicated information fails to improve the joint performance, re-planned paths and previous paths are the same. The reward for vehicle i choosing action a_j is

$$R\big(b_\tau, a_j\big) = \max_{\gamma^{[i][j]} \in \Omega_{\gamma^{[i][j]}}} R(b_\tau, a_j, \gamma^{[i][j]}) \tag{5.10}$$

We note that the value and the cost of communication is implicitly stated in (5.9). The value of communication is the improvement in the joint performance due to communication

$$B\big(\tilde{\gamma}^{[i]} \cup \tilde{\gamma}^{[j]} \cup \gamma_\star^{[1:k\setminus i,j]}\big) - B\big(\gamma_\star^{[1:k]}\big)$$

and, the cost of communication is the difference between vehicle i's expected performance when following its optimal path and its expected performance when taking the path $\gamma^{[i][j]}$ to maneuver towards vehicle j's communication range.

$$B\big(\gamma_\star^{[i]}\big) - B\big(\gamma^{[i][j]}\big)$$

It is also possible to add a penalty term for choosing a communication action to represent the computational effort of re-planning the paths.

Let Π be the set of all communication policies. Our objective is to find the optimal policy $\pi^\star$ that yields when and with whom to communicate

$$\pi^\star = \arg\max_{\pi \in \Pi} \mathbb{E}\Big[\sum_{\tau=0}^{\mathcal{N}} R(b_\tau, \pi(b_\tau))\Big] \tag{5.11}$$

Avoiding redundant computations of the value of communication when divergence from expected benefit is small

In the proposed solution, we compute the reward in (5.10) for choosing action a_j for every planning instance and for all $j = 1, 2, \ldots, k-1$. In practice, this can be very expensive and may lead to an intractable solution. Thus, we propose a simple and intuitive heuristic to determine when computing the reward (5.10) can be avoided.

We first note that if all vehicles perform as expected (e.g. expected risk reduction is attained in each visited cell), communication has no value. Indeed, communication may

improve the joint search performance only in two cases: 1) when a vehicle performs much worse than expected, and 2) when a vehicle performs much better than expected. In our search problem, the former represents the cases where the attained risk reduction along the path traversed by vehicle i turns out to be much lower than the expected risk reduction prior to acquiring any observation. In such cases, the expected utility of re-visiting the corresponding cells is likely to increase, and thus, the joint performance can be improved by assigning vehicle j to re-visit these cells. Similarly, the latter represents the cases where initially vehicle i is expected to perform poorly and vehicle j is assigned to search the same portion of the environment to attain a satisfactory performance, but during plan execution vehicle i attains sufficiently large risk reduction, and thus, assigning vehicle j to another portion of the environment may improve the joint performance. In both cases, in order to assign a different path for vehicle j, vehicle i communicates with vehicle j. By keeping track of how much the attained risk reduction along vehicle i's path has diverged from its expected value, we can determine when reasoning about communication with vehicle j can be avoided.

After computing the optimal paths $\gamma_\star^{[1]}, \gamma_\star^{[2]}, \ldots, \gamma_\star^{[k]}$ in plan-time, we also compute a threshold $\eta^{[i][j]}(q_n^{[i]})$ for each vehicle i and vehicle j and for each cell along vehicle i's path. We refer to this threshold as the *optimality-breaking* threshold, and it represents the least divergence from vehicle i's expected performance we should observe so that the value of communication can be greater than its cost. That is, if the divergence in vehicle i's performance up to cell $q_n^{[i]}$, the nth cell along vehicle i's path, is less than this threshold, following the same paths for the vehicles is still the optimal, and thus, computing the reward for communicating with vehicle j can be avoided. We compute the thresholds for all $i, j \leq k$ in plan-time and store them for comparison in run-time. The threshold $\eta^{[i][j]}(q_n^{[i]})$ is

$$\eta^{[i][j]}(q_n^{[i]}) = B\big(\gamma_\star^{[i]} \cup \gamma_\star^{[j]}\big) - B\big(\gamma_\star^{[i]} \cup \tilde{\gamma}^{[j]}\big) \tag{5.12}$$

where $\tilde{\gamma}^{[j]}$ is the optimal path for vehicle j that visits cell $q_n^{[i]}$.

Let $\Delta(i, k \mid z)$ denote the marginal benefit acquired from kth visit to cell i when z is previously observed

$$\Delta(i, k \mid z) = r(i, k \mid z) - r(i, k-1 \mid z) \tag{5.13}$$

where $r(i, k \mid z)$ is the expected risk for visiting cell i k times. After seeing measurement $z_{q_n^{[i]}}$ from each cell $q_n^{[i]}$, we compute the total amount of divergence from the expected benefit of a second visit to those cells

$$d\big(\gamma_\star^{[i]}(n)\big) = \sum_{q_l^{[i]} \in \gamma_\star^{[i]}(1:n)} \Big(\Delta(q_l^{[i]}, 1 \mid z_{q_l^{[i]}}) \; - \; \Delta(q_l^{[i]}, 2)\Big) \tag{5.14}$$

Computing the reward for taking action a_j to communicate with vehicle j can be avoided as long as

$$d\big(\gamma_\star^{[i]}(n)\big) \leq \eta^{[i][j]}(q_n^{[i]}) + C_\Sigma \tag{5.15}$$

where C_Σ is the cost of communication associated with the effort to transmit the information Σ and the effort to re-plan the paths. Our preliminary results show that this approach drastically reduces the number of times we compute the reward for a communication action, and it leads to tractable solutions for computing the communication policies.

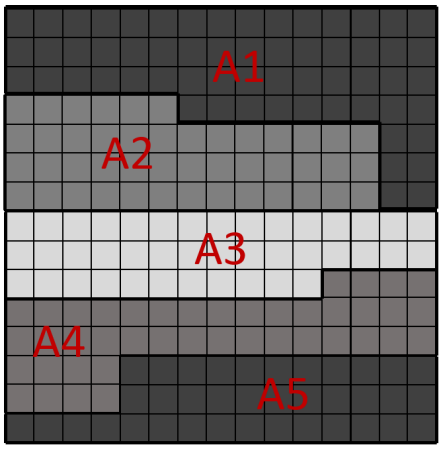
When computing the optimal communication policies, we currently assume that each vehicle deterministically knows where the other vehicles will be at any given time. Indeed, a more realistic assumption is to allow stochasticity in the other vehicles' locations. For instance, when vehicle i considers communicating with vehicle j, there may be non-zero probability that vehicle j has diverged from its pre-determined path $\gamma_\star^{[j]}$. However, this case is deferred for a future study.

5.4 Numerical results

In this section, we present the results of our numerical illustrations to show the efficacy of our approach for the multi-vehicle search problem. We consider simplistic scenarios where there are two search vehicles collaboratively searching the environment and the goal is to maximize the attained joint risk reduction after a mission. We assume that each vehicle has a sufficiently large communication range so that when a communication action is triggered, the vehicles can communicate with each other without changing their current paths. The cost of communication is mainly associated with the effort to re-plan the paths for both vehicles.

We note that while this is a simplified version of the problem described in Section 5.3, it still represents the challenges associated with a multi-agent planning problem.

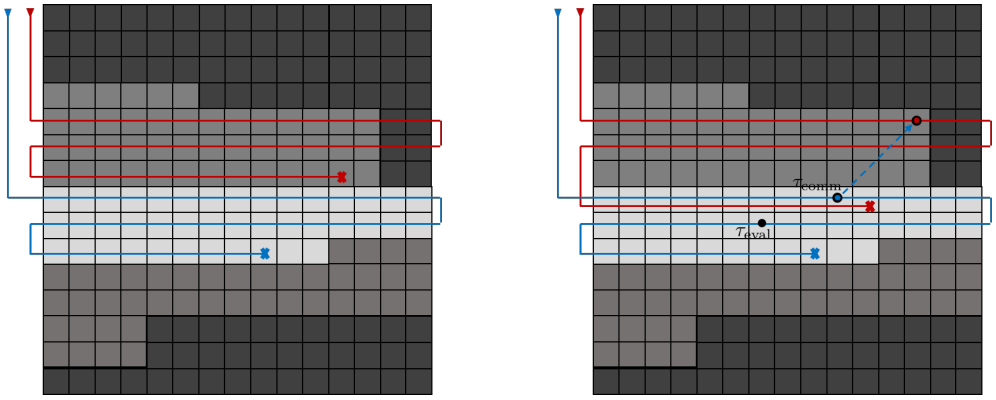
Numerical illustrations are performed on a 15 by 15 search environment. Fig. 5.2 shows the search area and the probability distribution for each cell. The search area is partitioned into regions A1 through A5. The probability of detection D and the probability of at least one false alarm F are shown in Table 2.1. As before, environment w_1 is the least and environment w_3 is the most informative. We consider that each vehicle has a mission length of 50.



Region	Probability distribution		
	w_1	w_2	w_3
A1	[0.95	0.05	0.00]
A2	[0.80	0.10	0.10]
A3	[0.20	0.70	0.10]
A4	[0.85	0.10	0.05]
A5	[0.90	0.10	0.00]

Figure 5.2: Search area and cell-wise environment distributions

In plan-time, the optimal paths for a planning horizon of 50 are computed for both vehicles by applying the exact branch-and-bound algorithm in Section 6.2.1. We also compute the optimality-breaking threshold in (5.12) for both vehicles and for each cell along the vehicles' paths in plan-time. Then, given the optimal paths and the optimality-breaking thresholds, each vehicle computes its optimal communication actions in run-time based on its local observations. At each planning instance, the vehicles reason about the cost and the value of communication and choose to communicate when the value of communication is greater than its cost. We consider a myopic communication planning approach to compute the optimal communication actions due to the large observation space ($|Z| = 51$). A possible extension of this work is to employ a sampling-based POMDP algorithm (see, for example, [71, 72]) to reduce the observation space so that a non-myopic communication planning approach can

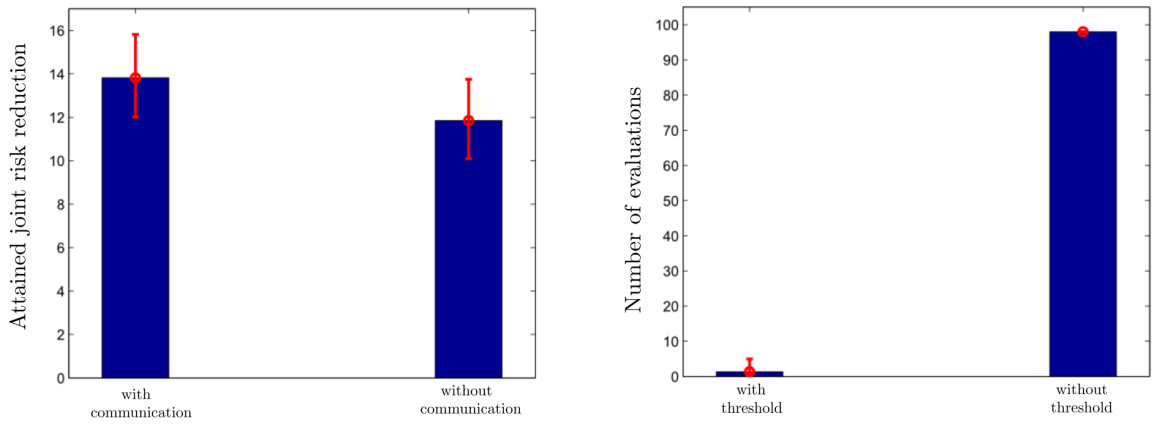


(a) (b)

Figure 5.3: A scenario where communication improves joint performance. Figures show a) the initial paths for both vehicles, and b) updated paths after communication occurs.

be applied. Fig. 5.3 shows a scenario where communication occurs and initial paths are updated to improve joint performance. The blue line and the red line represent the vehicle trajectories. Fig. 5.3a shows the initial paths for the vehicles computed in plan-time. As both vehicles execute their own paths, their actual performances differ from their expected performances. When the difference between the actual performance and the expected performance of a vehicle is larger than the corresponding optimality-breaking threshold, we compute the value of communication. In Fig. 5.3b, τ_{eval} and τ_{comm} denote the time a vehicle reasons about communication and a vehicle chooses to communicate, respectively. When a vehicle reasons about communication but it does not choose to communicate, this implies that the value of communication is expected to be less than the cost of communication. It is seen that after vehicles communicate at time τ_{comm}, the vehicle with red trajectory changes its path to sample from the same cells as the vehicle with the blue trajectory.

We conduct Monte Carlo simulations to assess the effect of communication on joint search performance and to assess the efficacy of the optimality-breaking threshold in Section 5.3.2 to reduce the computational complexity of the solution. For each cell in the search area, we randomly sample the true environment e from the environmental distributions in Fig. 5.2 and the true number of targets x from a uniform distribution. Then, we randomly generate search measurements z from the probability distribution $P(z \mid x, e)$ given true number of targets and true environment at a cell. When a vehicle visits a cell, it acquires the generated



(a) (b)

Figure 5.4: Comparisons of a) attained joint risk reduction after a mission, and b) the number of times the value of communication is computed

measurement z from that cell. Fig. 5.4 shows the results for 1000 iterations. In Fig. 5.4a, we compare the attained joint risk reduction with an error bar of 95% confidence interval for the cases where communication is available and where communication is not available. The difference between both cases is statistically significant ($p < 0.01$, Student's t-test), and the average joint risk reduction attained after a mission is improved by 12% due to communication. In Fig. 5.4b, we compare the number of times the value of communication is computed when the heuristic in Section 5.3.2 is employed and when it is not employed. When the heuristic is not employed, we simply compute the value of communication at every planning instance. Thus, for both vehicles, the value of communication is computed 98 times in total during a mission. On the other hand, Fig. 5.4b shows that when the proposed heuristic is employed, on average, the value of communication is computed only 2 times during a mission. While our approach of separating path planning from communication planning is a modest solution to multi-vehicle search problem, the results show promise to compute near-optimal plans for multiple vehicles in feasible time.

Chapter 6

Approximate Solutions for Search Problems

In this chapter, we address the computational challenge of computing the approximate search and environment characterization paths form the preceding chapters, and we show that the well-known approximation algorithms in the literature yield near-optimal paths in significantly less time compared to a brute-force approach. The results of this chapter are applicable to compute the optimal paths in Sections 2.2.1, 2.2.2, 3.4, and 3.5.2. In Section 6.1, we introduce two approximation approaches – a branch-and-bound algorithm and a Monte Carlo tree search algorithm – that are commonly employed in search applications. In Section 6.2, we apply the branch-and-bound approach to compute near-optimal paths over a small, notional, search area. Finally, in Section 6.3, we use a Monte Carlo tree search planner to compute near-optimal paths in a large search environment, and we test the efficacy of our approach using real sonar data previously acquired from Boston Harbor. In order to compute the value of a leaf node, we offer a novel heuristic for the Monte Carlo tree search planner that is tailored specifically to vehicles performing synthetic aperture sonar processing, such as in subsea applications. We acknowledge the help of our collaborators James McMahon, Artur Wolek and Zachary J. Waters from the U.S. Naval Research Laboratory and Nicholay Topin from Carnegie Mellon University Machine Learning Department for the findings of this chapter.

6.1 Approximation algorithms

Computing the optimal paths in Sections 2.2.1, 2.2.2, 3.4, and 3.5.2 requires enumeration of all candidate paths and evaluation of each path separately, which can be computationally infeasible when the planning problem has a large state-space. Thus, we are interested in approaches that reduce the computational requirements of computing optimal paths in Sections 2.2.1, 2.2.2, 3.4, and 3.5.2.

In the literature, two methods are commonly employed to reduce the computational complexity of a search problem. The first method is the branch-and-bound method which reduces the computational complexity by incrementally constructing the search tree and pruning the unpromising nodes of the tree when possible. And, the second method is the Monte Carlo tree search method where a search tree is progressively built based on random explorations of the search space according to a default simulation policy.

6.1.1 Branch-and-bound method

Branch-and-bound (BnB) algorithms are commonly used to solve large state-space search problems [103–106]. They repeatedly partition the problem into subproblems, which is called *branching*, and compute a lower and an upper bound on these subproblems, which is called *bounding*. Subproblems that will not lead to the optimal solution are pruned. This reduces the computational complexity of the problem compared to a brute-force search. Selecting the subproblems for branching is performed by using a systematic search strategy that satisfies the computational and the memory requirements of the problem. Commonly used search strategies are depth-first search (DFS), where the current node is explored to a specified depth until it is determined to be suboptimal, breadth-first search (BrFS), where neighboring nodes are expanded earlier than *deeper* nodes, and best-first search (BFS), where the most promising nodes, e.g. the nodes that have the largest upper bounds, are expanded first. The search is performed until the search space is fully explored, yielding the optimal solution. We refer the reader to a recent survey in [106] for more details.

The pseudo-code in Algorithm 1 outlines the steps in a general BnB algorithm. Without loss of generality, suppose the goal is to maximize a given objective function. Starting from

Algorithm 1 Branch-and-Bound Search

```
1: function BRANCH-AND-BOUND(v_0)
2:     Set L = {∅} and v = v_0, initialize UB = ∞, LB = 0, ÔPT
3:     while L is non-empty or v = v_0 do
4:         V_C ← GENERATECHILDREN(v)
5:         L ← REMOVE(v, L)
6:         for v' ∈ V_C do
7:             UB(v'), LB(v') ← COMPUTEBOUNDS(v')
8:             if UB(v') > LB then
9:                 L ← L ∪ v'
10:                UB, LB ← UPDATEBOUNDS(UB, LB, UB(v'), LB(v') )
11:                ÔPT ← UPDATESOLUTION(ÔPT)
12:        v ← select v' ∈ L
```

a root node v_0, a search tree is iteratively built. At each iteration, a node is selected from a set of candidate solutions and explored to a greater depth. In order to store the candidate solutions, an array L is used, and the search continues until L is empty. It is often assumed that an upper bound(**UB**) and a lower bound (**LB**) on the optimal solution is available. Otherwise, they are set to $\mathbf{UB} = \infty$ and $\mathbf{LB} = 0$. In addition, an estimate of the optimal solution $\hat{\mathbf{OPT}}$ (often called incumbent solution) is initialized to some initial value and it is updated when a greater solution is found during the tree search (Alg.1, Line 11). While L is not empty (Alg.1, Line 3), a node v is iteratively selected from L based on the employed search strategy (e.g. DFS, BrFS, BFS, etc.), its children nodes are generated (Alg.1, Line 4), and then the node is removed from L (Alg.1, Line 5). For each child node, an upper bound and a lower bound are computed for that node (Alg.1, Line 7). When the upper bound of the child node is greater than the lower bound **LB** on the solution, the child node is inserted into the set of candidate solutions (Alg.1, Line 9), and the upper bound and the lower bound on the solution are updated with the child node's upper bound and lower bound (Alg.1, Line 10). When the upper bound of the child node is not greater than the lower bound, the child node yields a suboptimal solution and thus it is pruned. Selecting a new node from the set of candidate solutions continues until all nodes are pruned. The algorithm terminates when L is empty (all candidate solutions are explored) and it returns the optimal solution $\mathbf{OPT} = \hat{\mathbf{OPT}}$. In general, the BnB algorithm has a worst-case complexity of $\mathcal{O}(\mathcal{N}^{|V_C|})$

where $\mathcal{N}$ is the planning horizon and $|V_C|$ is the number of children nodes generated at each step.

Computing the upper and the lower bound plays the key role to achieve the desired speed-up in computations. When the difference between the bounds is large, often fewer nodes are pruned, which slows down the computations. On the other hand, computing tight bounds such that the difference between the upper bound and the lower bound on a node will be small is often not possible due to lack of adequate domain knowledge or insufficient time to compute such bounds. Hence, there is a trade-off between the desired reduction in the size of the search tree and desired difference between the upper bound and the lower bound. A common approach to alleviate this problem is to compute approximate bounds that are easy to compute and sufficiently close to the true bounds. Computing efficient approximate bounds increases the efficacy of the branch-and-bound method and often allows real-time planning to achieve a near-optimal performance.

6.1.2 Monte Carlo tree search

Monte Carlo tree search (MCTS) is one of the most commonly employed methods to compute near-optimal decisions in large state-space problems. Given a start node, also called the root node, MCTS continuously builds an asymmetric tree of the search space with a best-first search strategy. The exploration of the tree nodes is guided by the random simulations that started from previously explored tree nodes. While MCTS has been mostly applied in combinatorial games such as the popular board games Go, [107], and Line of Actions (LOA), [108], it is also applied to a wide range of non-game applications [109–113]. One of the strengths of MCTS method is that it is an anytime algorithm. That is, the newly acquired information is backpropagated to the root node immediately so that the current best decision is readily available at any time. We refer the reader to a recent survey paper in [114] for more details.

Algorithm 2 outlines the steps of the anytime MCTS planner. At each move, the planner incrementally builds the search tree by recursively expanding leaf nodes from the current root node v_0 (Alg.2, line 6) until the allowed time per move is reached and an action is requested (Alg.2, line 5). When the allowed time is reached, the algorithm returns with the

current best action based on the action selection criteria (Alg.2, line 7). Then, the selected best action is executed and the resulting node is assigned as the current root node.

Algorithm 2 Monte Carlo Tree Search

```
1: function ONLINE-MCTS(s_o)
2:     create root node v_0 with state s_0
3:     initialize expected reward v_{0.E} = 0
4:     while mission length ≥ 0 do
5:         while action not requested do
6:             v_{0.E} ← MCTS(v_0)
7:         a ← BESTACTION(v_0)
8:         v_0 ← STEP(v_0, a)
```

Algorithm 3 outlines a general MCTS framework to gradually build the search tree. It consists of 4 steps: selection, expansion, simulation and backpropagation. In a nonterminal state, the best child of the current internal node is selected recursively (Alg.3, lines 8-9) until an unexpanded node is found. The value N_v in Alg.3, line 5 is the visit count associated with a node v. Often, the well-known UCT (Upper Confidence Bound 1 applied to trees) algorithm in [111, 115] is employed to address the trade-off between exploiting the current best node and exploring other less frequently selected nodes, which enables an asymmetric growth of the tree where more promising nodes are explored to a greater depth. When an unexpanded node is encountered, its children nodes are created (Alg.3, line 4). A default policy is used to simulate the game-theoretic value of the selected node when the selected node is being visited for the first time (Alg.3, line 6). After running the default policy, the returned value of the simulation is backpropagated recursively all the way up to the root node and the visit count for each internal node along the way is incremented by 1 (Alg.3, lines 10-11). When a terminal state is reached (i.e., when the planning horizon is satisfied), the algorithm returns with its accrued reward. Given that the number of simulations from a node is sufficiently large, the estimated game-theoretic value of that node is expected to accurately represent its true value.

6.2 Applying branch-and-bound approach to search problem

We now apply the branch-and-bound (BnB) approach from Section 6.1.1 to the search problem described in Section 3.4.2. We consider an exact BnB planner and an approximate BnB planner, and we show that applying a BnB planning approach to the search problem yields near-optimal results within feasible time. The approximate BnB algorithm is proposed by James McMahon from the U.S. Naval Research Laboratory, and it is presented here for comparison and completeness.

We first note that the expected reward acquired for visiting a cell multiple times is dominated by the expected reward acquired from the first visit to that cell. Thus, we modify the problem in a way that only the first visit to a cell incurs a non-zero reward and further visits to that cell has no reward. This enables us to cast the search problem as the well-known orienteering problem where the search vehicle collects a one-time reward for visiting a specific location and the goal is to maximize the collected reward within limited time/distance the vehicle can travel.

Algorithm 3 Monte Carlo Tree Search

```
1:  function MCTS(v)
2:      if v is nonterminal then
3:          if v has no children then
4:              Expand(v)
5:          if N_v = 0 then
6:              Δ ← DefaultPolicy(s(v))
7:          else
8:              v' ← SelectBestChild(v)
9:              Δ ← MCTS(v')
10:         N_v = N_v + 1
11:         v.𝔼 = max(v.𝔼, Δ)
12:         return v.𝔼
13:     else
14:         return EvalTerminalState(s(v))
```

6.2.1 Exact branch-and-bound planner

We first apply an exact branch-and-bound algorithm with a best-first search method to compute the optimal locations. While this method yields the optimal performance, it is computationally less efficient and real-time generation of optimal paths is not feasible when the planning horizon is large. We briefly discuss how we compute the lower bound **LB** and the upper bound **UB** in Algorithm 1.

Lower Bound

The lower bound **LB** used in the exact solution is calculated by summing the reward for searching a location over a lawn-mover path, a path that consists of parallel linear tracks until the mission length is met. Since this path belongs to the set of feasible paths, the optimal solution is guaranteed to yield a reward greater than or equal to the value of this path. This value is calculated once at the start of the search in $\mathcal{O}(K)$ time where K is the number of cells. When the value of expanded nodes of a candidate path exceeds this lower bound, we update the lower bound to be the value of this candidate path.

Upper Bound

To compute the upper bound **UB** for the maximum attainable reward along a given path, the maximum reward in a cell throughout the search area is multiplied by the number of cells the vehicle can visit without exceeding the mission length. As the search is expanded, the upper bound is re-computed by subtracting the maximum reward in a cell from the value **UB** and adding the reward associated with the newly added cell. The initial upper bound computation takes $\mathcal{O}(K)$ time while updates are done in $\mathcal{O}(1)$. While these bounds may not prune the search tree as effectively as tighter bounds, they take significantly less time to compute.

6.2.2 Approximate branch-and-bound planner

We also apply an approximate branch-and-bound algorithm to compute the near-optimal paths in real-time. The approximate algorithm is proposed by James McMahon in [116],

and it is presented here for comparison. The planning problem is first re-formulated as a search over a graph where the vertices of the graph $v_i \in \mathcal{V}$ correspond to horizontal rows over the search area $\mathcal{G}$, and the edges $\epsilon_{ij} \in \mathcal{E}$ are the distance required to travel between rows. Then, the search problem is defined over this graph as follows:

Definition 1. Given

- An undirected symmetric graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,
- A maximum tour length of $\mathcal{N}$,
- A reward, R_i associated with each vertex $v_i \in \mathcal{V}$, and
- A cost, $c_{i,j}$ associated with each edge $\epsilon_{i,j} \in \mathcal{E}$,

compute an ordered open tour $T^* = \{v_1, v_2, ..., v_h\}$ that visits a subset of vertices $v \subseteq \mathcal{V}$ such that

$$v_i \neq v_j \quad \text{for } i, j \in \{1, ..., h\},\ i \neq j, \tag{6.1}$$

$$v_1 \in \mathcal{T}^*, \tag{6.2}$$

$$\sum_{i=1}^{h-1} c_{i,i+1} \leq \mathcal{N}, \tag{6.3}$$

$$\sum_{i=1}^{h} R_i \quad \text{is maximized.} \tag{6.4}$$

This problem definition follows that of the *Selective Traveling Salesman Problem* or *Orienteering Problem* which is a special case of the *Traveling Salesman Problem* and has been heavily studied in the literature (see [117,118] for a comprehensive survey). The reward R_i is a function of the sum of the rewards across the entire row a survey would take place in.

Following the upper bound on the optimal solution defined in [119], a slight modification is made such that the tour is not constrained to end at a specific node. Following the

approach in [120], upper bounds are computed by searching for a terminal vertex in a sorted array of weighted vertices. That is, given a sorted array of non-increasing values of the reward per unit weight:

$$R_j/\bar{c}_j \geq R_{j+1}/\bar{c}_{j+1} \quad \text{for} \quad j = 1, \ldots, n_r - 1 \tag{6.5}$$

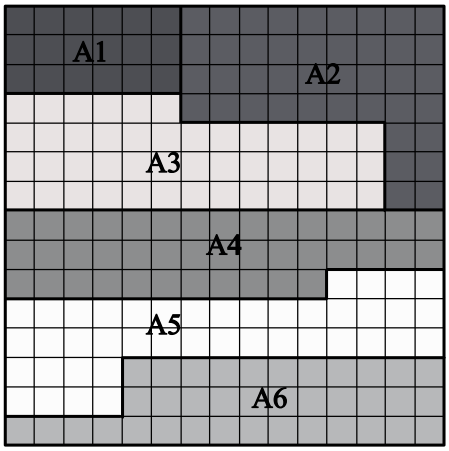
where $\bar{c}_j = \arg\min_{i \neq j}(c_{i,j})$ for $j = 1, \ldots, n$. The upper bound on the reward can be computed by finding the maximum reward that can be obtained without exceeding the budget $\mathcal{N}$. Since this is a summation over the sorted array, the upper bound has a complexity of $\mathcal{O}(n_r)$. Further detail on how to tighten the bounds can be found in [120].

Using these upper bounds, solutions for the planning problem are found by implementing a depth-first branch-and-bound (DFBnB) solver [121] where each node in the search tree corresponds to performing different survey lines. The branching of the tree is ordered such that nodes associated with higher reward per weight (Eq. 6.5) are evaluated first. To prune the search based off of the maximum possible reward at each node, the upper bound described in [119] is employed. This bound can be re-computed during each iteration in $\mathcal{O}(n_r)$. Furthermore, each time a new tour is discovered that yields a higher reward, the ordering of the tour is optimized via 2-opt [122] in order to ensure the lowest cost tour for the highest reward is being produced. Instead of performing the search exhaustively, after a large number of iterations the search is terminated.

6.2.3 Numerical results

We provide the results of the numerical illustrations to emphasize the applicability of approximate methods in search problems. For a detailed description of the search problem and how the numerical results are obtained, we refer the reader to [116]. The approaches, exact BnB method and approximate BnB method, are compared in terms of overall search performance and computation time, over a small, notional, search area (Fig. 6.1). The corresponding C++ code is implemented by James McMahon and Artur Wolek from the U.S.

Naval Research Laboratory along with the author of this dissertation.



Region	Probability distribution		
	w_1	w_2	w_3
A1	[0.95	0.05	0.00]
A2	[0.85	0.10	0.05]
A3	[0.10	0.10	0.80]
A4	[0.30	0.40	0.30]
A5	[0.00	0.10	0.90]
A6	[0.10	0.65	0.25]

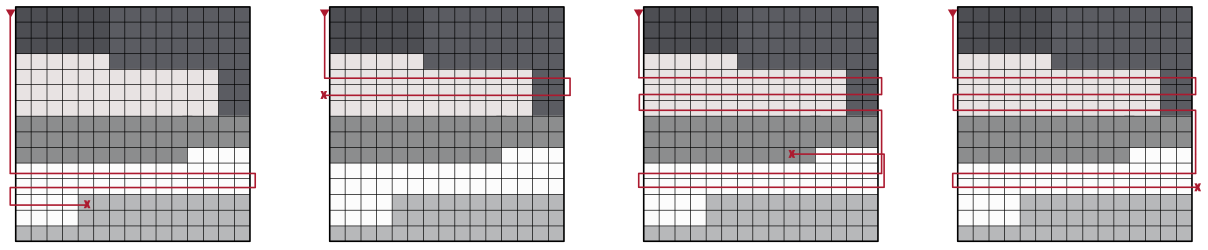
Figure 6.1: Search area and cell-wise environment distributions

The search problem that we consider for the numerical illustrations follows from Section 3.4. The search vehicle is equipped with a search sensor and an environment characterization sensor and both sensors operate simultaneously. We consider that the objective of the search mission is to minimize the risk associated with incorrect estimation of number of targets. Thus, the value of searching a location is computed from (3.32), and the objective is to maximize (3.34). For notational convenience, let $B(i)$ denote the value of searching cell $i \in \mathcal{G}$. That is,

$$B(i) = \sum_{z_i} \sum_{y_i} P(z_i, y_i) \sum_{x_i} P(X_i = x_i \mid z_i, w_j) L_X(x_i, \delta_X(z_i)) \tag{6.6}$$

The probability of detection, D, and the probability of at least one false alarm, F, for each environment are shown in Table 2.1. We use the sensor model in (3.51) for environment characterization with $a_{11} = 0.82$, $a_{22} = 0.84$, $a_{33} = 0.88$, and $a_{ij} = a_{ik}$ for $j, k \neq i$. The comparison between the different planners was conducted over a 15-by-15 cell search area with known cell-wise environment distributions as shown in Fig. 4.1. The search area is divided into 6 regions: A1 through A6. For each region, the corresponding probability distribution $\Pi = [p_1, p_2, p_3]$ is given, where p_j is the probability that the environment is w_j.

Fig. 6.2 shows the search paths for both algorithms for a mission length of 50 (Fig. 6.2a-b)



(a) (b) (c) (d)

Figure 6.2: Search paths for a) exact-solution method b) approximate solution method when mission length is 50 and c) exact-solution method d) approximate solution method when mission length is 100.

and for a mission length of 100 (Fig. 6.2c-d). The path traversed by the vehicle is represented by the red line. The averaged normalized achieved risk reduction (Eq. 6.7) associated with the corresponding mission length is 7% for the approximate planner (Fig. 6.2b) and 12% for the exact planner (Fig. 6.2a) when mission length is 50, and 21% for the approximate planner (Fig. 6.2d) and 23% for the exact planner (Fig. 6.2c) when mission length is 100. The results show that while the exact method yields the optimal search path, the approximate method results in a near-optimal search path. For both methods, the vehicle skips the parts of the search area where expected risk reduction is relatively small. We note that since the approximate method considers each row as a vertex, it does not start a row if it will not finish it. Thus, when using the approximate method, search may stop before the mission length is met.

Based on the environmental distributions in Fig. 4.1, a total of 500 synthetic environments were generated for a particular mission length. For each synthetic environment, a corresponding set of measurements was generated based on the prior belief on the number of targets and the environmental conditions. The mission length (in terms of number of cells) was varied in increments of 10 from 50 to 110. The synthetic environment, the measurements, and a specified mission length were then passed to both solvers (corresponding to the exact and approximate algorithms) consecutively. At each iteration the respective solvers recorded the sequences of planned paths, the computation time, and the search performance.

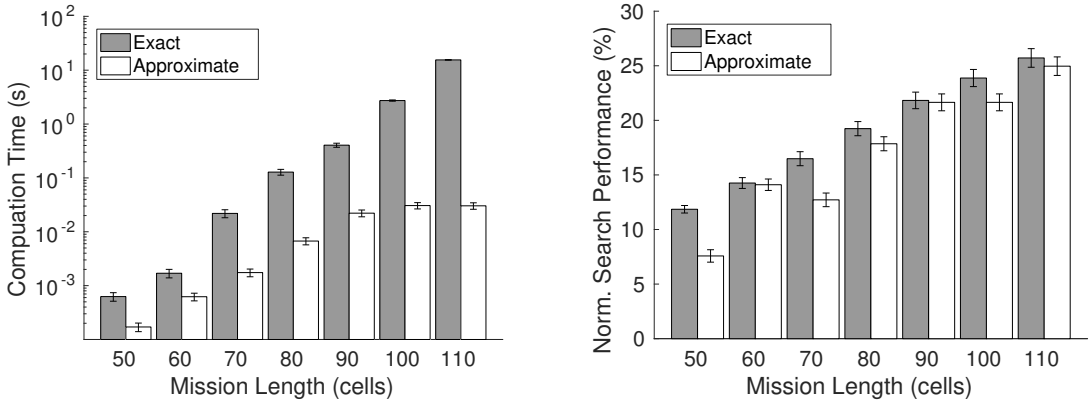


Figure 6.3: Comparison of average computation time (a) and average normalized search performance (b) between exact and approximate algorithms based on Monte Carlo simulation. Error bars indicate one standard deviation. The figure is created by Artur Wolek

The algorithms were implemented in C++ using the Armadillo linear algebra library [123] on a 64-bit Ubuntu 16.04 operation system. All computations were executed on an Intel Xeon E5-1650V3 processor with a processor base frequency of 3.5 GHz and 32 GB of RAM.

The results of the Monte Carlo simulation are shown in Fig. 6.3. As expected, the search performance and computation time increased with mission length for both algorithms. We found that, in general, the approximate algorithm had comparable search performance to the exact algorithm with a substantially lower computation time. The difference in computation time is particularly pronounced as the mission length increases (e.g., for a mission length of 100 cells the difference is about two orders of magnitude). We note that in Fig. 6.3, achieved risk reduction after a mission is normalized with the risk prior to acquiring any measurement. That is, let γ_j be the path the vehicle traverses at the jth run, the averaged normalized achieved risk reduction $\bar{B}_{\text{avg}}$ is

$$\bar{B}_{\text{avg}} = \frac{1}{\mathcal{M}} \sum_{j=1}^{\mathcal{M}} \left(\frac{\sum_{i \in \gamma_j} B(i)}{\sum_{i \in \mathcal{G}} \rho_i} \right) \tag{6.7}$$

where $\mathcal{M}$ is the number of Monte Carlo iterations, and ρ_i is the prior risk in cell i before acquiring any measurement. Thus, Fig. 6.3 gives the percentage of the achieved risk reduction over a long run (e.g., on average, both approaches yield above 20% risk reduction throughout

the search area).

6.3 Applying Monte Carlo tree search approach to search problem

In Section 6.2, we address the computational challenge of computing near-optimal search paths and develop an exact solver and an approximate solver based on a special case of the traveling salesman problem. The approximate solver yields comparable results with the exact solution while it requires significantly less computational power. However, in Section 6.2, we consider that the search sensor has the typical characteristics of a side-scan sonar, and thus, the vehicle is required to travel in straight lines across the entirety of the map. In this section, we extend our results in Section 6.2 by allowing the vehicle to make turns within the map as opposed to taking parallel straight lines (only). We note that this is not a small extension due to the substantial increase in problem complexity, and the approach reported herein is fundamentally different than our approach in Section 6.2.

To account for turns within the map we add a penalty to the cost function each time a turn action is taken. The numeric value of the penalty is computed off-line by considering the additional time required to maneuver to an adjacent cell assuming a kinematic vehicle model. That is, either the vehicle goes straight or makes a turn to move into a more promising part of the search area at the expense of paying a penalty for making a turn. While this theoretically results in a greater or no worse performance than the case where the vehicle is required to travel in a straight line, the described problem has a drastically larger state space that requires significantly greater computational power. In order to meet the computational requirements of the problem, we employ a Monte Carlo tree search (MCTS) planner. Often, MCTS planners consider a random walk of the discretized search space to determine the value of a leaf node. However, in our search problem, random walk can result in poor search performance since the turn penalties accrue over large planning horizons but only impact reward towards the end of planning (i.e., they influence the effective mission length). Thus, we also propose two novel heuristics to determine the value of a leaf node. We test the efficacy of our heuristics through extensive numerical simulations where search paths are

planned using sonar data previously acquired from Boston Harbor. Our results show that both heuristics perform significantly better than a random walk, and the second heuristic yields a greater performance increase over all other methods when turn penalty is high.

6.3.1 Vehicle model and action costs

We consider that the search vehicle is performing information gathering using a dual-sided synthetic aperture sonar (SAS) [41, 42]. SAS processing requires ensonifying the same area on the sea-floor from multiple positions along a straight-line track. Therefore, to effectively gather information from a cell, data collection must begin at some point before the vehicle enters the cell (i.e., during the *lead-in*) and terminate at some point after it exits the cell (i.e., during the *lead-out*). The lead-in length L_i after a turning motion is often chosen larger than the lead-out length L_o to allow any transient motions to dampen before data collection begins.

When planning a survey route over the search grid the vehicle has the option to perform an action $a \in \mathcal{A} = \{a_1, a_2, a_3\}$ at each cell. The available actions are: *move forward* (a_1), *turn left* (a_2), or *turn right* (a_3). We specifically assume that the cost of each action is equal to the length of the corresponding path. A *move forward* action causes the vehicle to continue traveling along the centerline of a row or column of cells. During consecutive *move forward* actions the lead-in/lead-out constraints need not be considered and the cost of each action is equal to the cell width L.

Turn actions require that the lead-in and lead-out segments be accounted for appropriately. A *turn left* (or *turn right*) action causes the vehicle to travel to a nearby cell with a 90 degree heading change. During turn actions we assume that SAS processing will not be effective and no information is gathered. To compute the cost of a turn, we assume the vehicle's motion satisfies a *Dubins* kinematic model [124]. The vehicle is represented with a state vector $\mathbf{x} = (p_x, p_y, \theta)$ where $(p_x, p_y) \in \mathbb{R}$ is the planar position and $\theta \in [0, 2\pi)$ is the heading. Let the vehicle's state at the end of the lead-out segment be $\mathbf{x}_o$ and the vehicle's state at the beginning of the lead-in segment be $\mathbf{x}_i$. Denote the cost of the minimum-length path (i.e., the Dubins path) that joins these two states as $D(\mathbf{x}_o, \mathbf{x}_i)$ assuming a minimum turning radius of R. We can now define the turn penalty constant.

Definition 2. The turn penalty $\tau = L_o + D(\mathbf{x}_o, \mathbf{x}_i) + L_i$ is the total length of the path that the vehicle follows when performing a turn action.

6.3.2 Random-lines heuristic policy

The first heuristic policy that we propose is the RANDOM-LINES heuristic. This heuristic is tailored specifically to vehicles performing synthetic aperture sonar processing, such as in subsea applications. The main idea of the RANDOM-LINES heuristic is to avoid taking a turn action unless the remaining part of the straight line is expected to yield relatively poor search performance so that the number of vehicle turns can remain sufficiently small while the segments of the search area with low rewards are omitted.

An initial step in the RANDOM-LINES heuristic is to determine where to make a turn in a line (a line can be a row or a column). That is, when the vehicle starts moving along a line, we want to know the number of cells for which the vehicle should move forward before making a turn. We compute this for each row and column in the search grid for available directions (East and West directions for rows, and North and South directions for columns) prior to building the search tree (prior to Alg.2, line 4), and store the resulting cells in the memory for later use when running the heuristic. To determine where to make a turn in a line, we compute the cost and the benefit of making a turn at each cell along that line. That yields a turn gain for each cell. Then, the cell with highest turn gain is selected as the cell to make a turn when sweeping the corresponding line during a simulation.

Let i be the index of the line for which we compute the turn gains, d be the direction for the vehicle (East, West, North or South), $l(i, j, d)$ be a function that returns the index of the jth cell of ith line along the direction d, and n_L denote either the number of rows (n_r) or the number of columns (n_c) depending on the corresponding line being a row or a column. Then, the turn gain for the jth cell is

$$g(i, j, d) = w_{i,d}\, b(i, j, d) - c(i, j, d) \tag{6.8}$$

where $b(i, j, d)$ represents the benefit for turning at the jth cell of line i in the direction d and it is the number of remaining cells along the line that the vehicle does not traverse

$$b(i,j,d) = n_L - j, \tag{6.9}$$

and $c(i,j,d)$ represents the cost for turning, and it is the ratio of the total reward of remaining cells in a line to the total reward along that line

$$c(i,j,d) = \frac{\sum_{k=j+1}^{n_L} B\big(l(i,k,d)\big)}{\sum_{k=1}^{n_L} B\big(l(i,k,d)\big)}. \tag{6.10}$$

where $B\big(l(i,k,d)\big)$ is the value of searching the kth cell of line i in the direction d, and it follows from (6.6).

We note that when computing the benefit and the cost of a turn in (6.9) and (6.10), we assume the vehicle starts line i from the first cell in direction d. Indeed, it is possible to start a line from a middle cell during a simulation, but assuming that a line will be started from its first cell results in fewer turns in general. The weight $w_{i,d}$ in (6.8) is the relative weight of the benefit of turning at a cell to its cost for the ith line along the direction d

$$w_{i,d} = \frac{1}{1 + w_\tau \ \tau\frac{\mathcal{N}}{K}} \times \frac{\frac{1}{2}\big(\max_k B(k) + \frac{1}{K}\sum_{k=1}^{K} B(k)\big)}{\sum_{j=1}^{n_L} B\big(l(i,j,d)\big)} \tag{6.11}$$

where $\mathcal{N}$ is the mission length, K is the number of cell in the search grid, τ is the turn penalty defined in Definition 2, and $w_\tau > 0$ is a tunable parameter. In (6.11), the denominator of the first term accounts for the effect of the turn penalty and also the ratio of the mission length to grid size, the numerator of the second term accounts for the average of the mean reward and the maximum attainable reward in the grid, and the denominator of the second term accounts for the total reward along a line. Informally speaking, the average of the mean reward and the maximum attainable reward in the grid represents an optimistic estimate of the reward the vehicle would acquire if it made a turn. Thus, making a turn is preferred if the next cell's reward is adequately smaller than this estimate. There is a trade-off between the turn penalty and the expected increase in the reward for making a turn, e.g. for a large turn penalty, the value of making a turn is smaller. We make a turn at the cell that maximizes the turn gain for the ith line along the direction d

Algorithm 4 Random Lines MonteCarlo Search

```
1: function HeuristicPolicy(s)
2:     Φ ← SelectSurveyDir(s)
3:     (i, j, d) ← GetCurrentLoc(s)
4:     if d ⊥ Φ then
5:         n ← ψ(i, d) − j
6:     else
7:         n ← rand([0, n_L'])                ▷ n_L' is the number of remaining cells along the line
8:     s, j ← Survey(s, i, d, n)              ▷ move along line i n steps in direction d
9:     while ¬ isTerminal(s) do
10:        i, d, Φ ← SelectLineAndDir(s, i, d, Φ)
11:        n ← ψ(i, d) − j
12:        s, j ← Survey(s, i, d, n)
13:    return EvalTerminalState(s)
14: function SelectLineAndDir(s, i, d, Φ)
15:    a ← ChooseTurnAction(i, d, Φ)
16:    if isValid(a) then
17:        n_L' ← turn(a)
18:        i' ← rand([0, 1]) × rand([0, n_L' − 1]) + 1
19:        i ← MoveToLine(i, a, i')
20:        d ← SelectDir(i)
21:    else
22:        Φ ← ReverseSurveyDirection(Φ)
23:    return i, d, Φ
```

$$\psi(i,d) = \max_{j} g(i,j,d). \tag{6.12}$$

Algorithm 4 shows the pseudo-code for the RANDOM-LINES heuristic. The first step of the heuristic is to randomly select a survey direction from the cardinal directions (East, West, South and North) to survey the search area (Alg.4, line 2). Then, given the current line i, the current direction d, and the current location j along line i in direction d (Alg.4, line 3), the heuristic selects the number of cells, denoted by n in the pseudo-code, that the vehicle will move forward before making a turn (Alg.4, lines 4-7). After moving along line i for n cells in direction d (Alg.4, line 8), the heuristic selects another line parallel to line i to sweep (Alg.4, line 10). With 0.5 probability we select the next parallel line, and with

remaining 0.5 probability we select one of the remaining lines after taking a turn action a (Alg.4, line 18). After selecting a new line and a direction, we move along that line in that direction for n cells (Alg.4, line 12). The heuristic re-iterates through the lines 10-12 of Algorithm 4 until a terminal state is reached.

We note that since a full-length candidate path is generated at each simulation, the RANDOM-LINES heuristic allows us to keep track of the current best path. Since the cell rewards are deterministic, the reward of the current best path can be a lower bound on the reward we attain after a mission if we choose to follow that path until the current best path changes. Thus, we choose an action selection criteria to strictly follow the current best path. We update the best path whenever a new candidate path with a greater terminal reward is simulated. Following the same strategy, when back-propagating the value of a leaf node to the root node (Alg.3, line 10), we return the maximum value of any simulation from that node. This strategy greatly improves the search performance when the exploration-exploitation parameter in UCT algorithm is tuned accordingly. When this parameter deviates from its optimal, the performance degrades gracefully, making it still a good strategy without any domain knowledge (see Fig. 6.5).

6.3.3 Voxelized state heuristic policy

The second heuristic policy, referred to as the VOXEL heuristic, is proposed by Nicholay Topin in [125], and it is presented here for comparison. The VOXEL heuristic aims to approximate the reward gained when surveying a subset of contiguous regions chosen from a set of voxels formed by overlaying a low resolution grid onto the map. Instead of performing a true simulation over the high resolution map, a policy designed to survey the multiple voxelized regions is used to estimate the reward.

First, the map is pre-processed into a set of *voxels*. Initially the resolution of the voxelized map is determined by using a tunable step parameter γ which corresponds to the number of steps required to survey the voxelized map given the remaining steps over the true map. Next, the heuristic policy performs a number of simulations where it recursively selects neighboring voxels to traverse based off of the current voxelized state. Valid states are defined by the unexplored neighboring voxels (i.e., North, East, West, and South) that are accessible to the

vehicle. Each iteration the heuristic determines the set of feasible neighbors, and if there exists at least one, it randomly selects one for expansion. Additionally, a modification to Alg. 3 line 8 is made to improve the expansion of the search tree. While the VOXEL heuristic does perform better than a pure random walk, it does suffer from similar issues in that it acts myopically with respect to the turn penalty. One way to improve this is to introduce an *action-biased* selection strategy for expanding the tree. Thus, the expansion is biased towards the action a_1 (move forward) such that the UCT becomes

$$UCT = \begin{cases} \mathbb{E}_j + 2C_\alpha \sqrt{\frac{2\ln n}{n_j}}, & \text{if } a = a_1 \\ \\ \mathbb{E}_j + 2C_\beta \sqrt{\frac{2\ln n}{n_j}}, & else \end{cases}$$

where $C_\alpha \geq C_\beta$ in order to bias towards action a_1.

6.3.4 Numerical results

In this section, we examine through numerical simulations the efficacy of the proposed MCTS heuristics in improving search performance. We specifically compare the performances of the RANDOM-LINES heuristic from Section 6.3.2 and the VOXEL heuristic from Section 6.3.3 with the performances of the approximate solver from Section 6.2.2, the mowing-the-lawn path and the commonly used random heuristic that implements a random walk of the search space. The corresponding C++ code is implemented by James McMahon and Artur Wolek from the U.S. Naval Research Laboratory and by Nicholay Topic from Carnegie Mellon University Machine Learning Department along with the author of this dissertation.

The trials are conducted over a search area consisting of 61 rows and 61 columns, a total of 3721 cells. There are three environments in the search area and for each cell a probability distribution over the environments is known. The environment probability distributions are obtained from the sonar data that was previously extracted from Boston Harbor using the Reliant AUV. The abstraction of the sonar data was performed by Zackary J. Waters from the U.S. Naval Research Laboratory, and it is discussed in detail in prior work [116]. The search area and the normalized cell rewards are shown in Fig. 6.4. The parts with blue

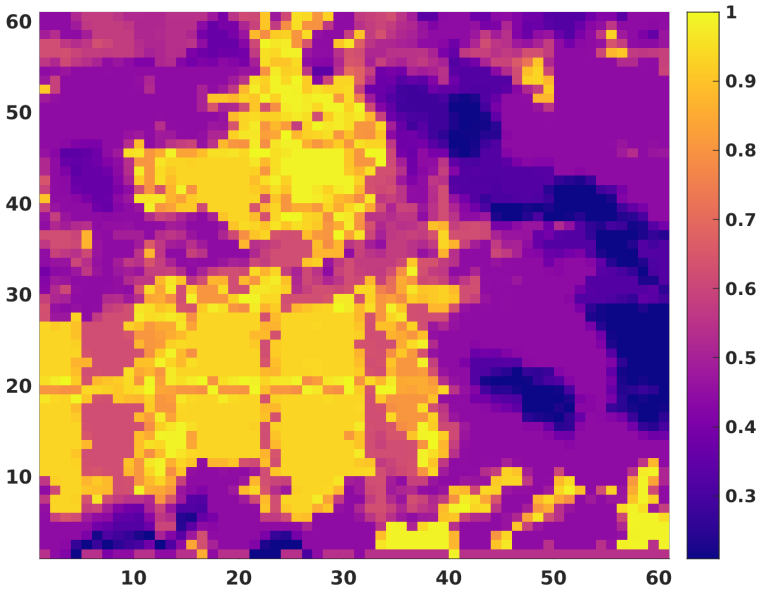
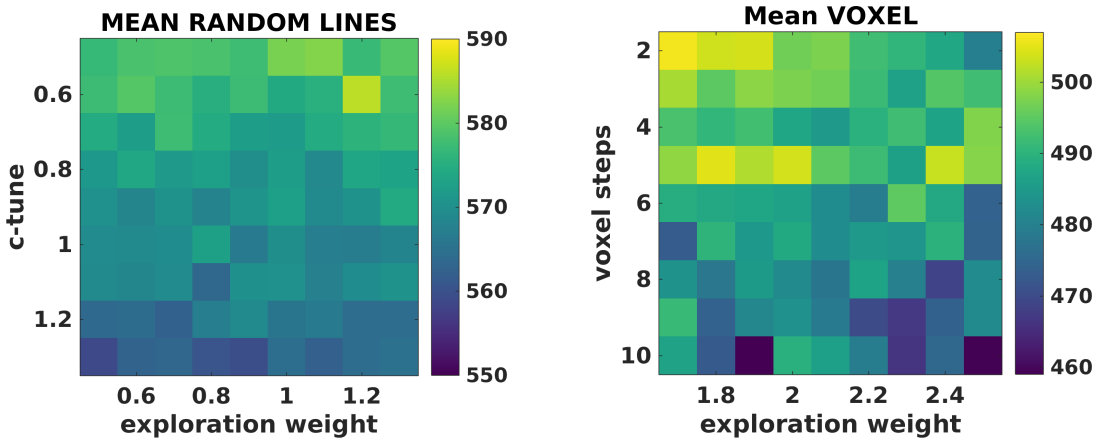


Figure 6.4: A subsea search environment showing an unpublished data set from the Boston Harbor taken in 2016. The x and y axis represent unit-less dimensions corresponding to the number of cells in the image (61×61). Cell-wise rewards are normalized to 1. The figure is created by James McMahon

color indicate poor search performance and the parts in yellow color indicate good search performance. We performed the comparison with several mission lengths (in terms of the number of traversed cells) varying from 500 to 2000 with an increment of 500. The allowed time per move was determined to be 20 seconds[1]. Thus, each trial took approximately 3 to 11 hours. Due to the lengthy simulation time required for a single trial we limited our analysis to include 40 trials for each mission length and each method using the MCTS algorithm. The results from the lawnmower approach as well as the approximate solver are the mean of two simulations, horizontal and vertical search (*North-South, West-East*) since the survey direction has a large impact on the baseline performance. The algorithms were implemented in C++ on a 64-bit Ubuntu 16.04 operating system using a single thread. Parallelization was used to take advantage of multiple threads (one thread per simulation) using GNU parallel [126]. For these experiments we performed an initial parameter study for a mission length of 1000 and selected the values $w_r = 0.6$ and $C = 1.2$ for the RANDOM-LINES heuristic and

[1]20 seconds corresponds to the amount of time it takes for the vehicle to travel through a cell. In practice the vehicle would update the trajectory every time it sampled a cell. When performing a turn action it plans for the duration of the turn, 60 seconds



(a) (b)

Figure 6.5: Parameter study of the (a) RANDOM-LINES and (b) VOXEL heuristics for a mission length of 1000. Each data point represents the reward averaged over 8 simulations. The figure is created by James McMahon

$\gamma = 5$ and $C_\alpha = 1.8$ for the VOXEL heuristic. Fig.6.5 shows the average results for various values of these parameters. We note that both heuristics yield acceptable performance even when these parameters are not tuned properly.

Fig. 6.6 compares the mission length and the attained reward (i.e., the attained risk reduction in (6.6) after a mission) for five different planners: the standard mowing-the-lawn path, the MCTS VOXEL (Section 6.3.3), the MCTS RANDOM-LINES (Section 6.3.2), the MCTS random walk, and the approximate planner (BnB) from Section 6.2.2. For each planner, the average reward and the standard deviation of 40 trials are shown. Both VOXEL heuristic and RANDOM-LINES heuristic yield significant increase in search performance over the random walk of the search space. In addition, the deviation from the average result is much lower in the proposed heuristics compared to other planners. This suggests that 1) MCTS is a promising approach in large-scale search missions, and 2) the proposed heuristics can be more useful than a random heuristic to determine the value of a leaf node during a tree search. The best performance over the series of experiments is obtained using the MCTS algorithm with RANDOM-LINES heuristic. This is especially evident in the shorter time horizons. It can be seen however, that the approximate solver from Section 6.2.2 yields similar performance as the horizon begins to increase, however the standard deviation of the

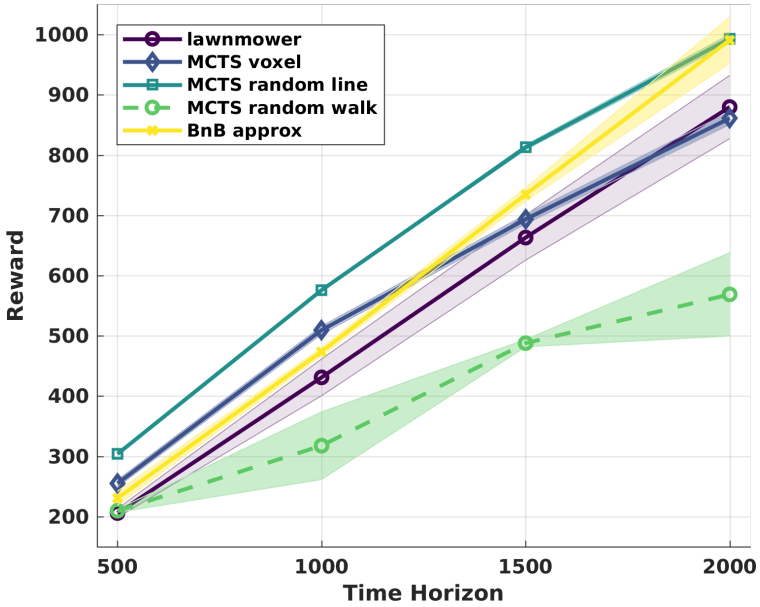
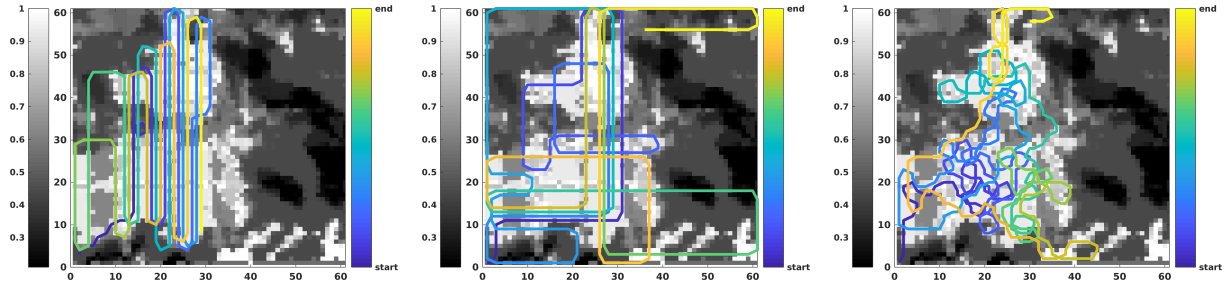


Figure 6.6: Attained reward of the various planners as a function of the mission length.The lines represent the mean value and the standard deviation (shaded regions) over the set of experiments. The figure is created by James McMahon



(a) (b) (c)

Figure 6.7: Traversed paths for a mission length of 1000 when RANDOM-LINES heuristic is employed (a), when VOXEL heuristic is employed (b), and when a random walk heuristic is employed. The trajectories begin at blue (bottom left of map) and end at yellow. The figure is created by James McMahon

solution grows. This is due to the distribution of reward through the environment, if large swaths of high-reward regions are grouped on one side of the map, the approximate solver yields a high-reward solution with increasing mission length since the corresponding optimal path will more closely resemble a straightforward lawnmower pattern.

Fig. 6.7a-c show the traversed path of a particular trial for a mission length of 1000 when RANDOM-LINES heuristic is employed (Fig. 6.7a), when VOXEL heuristic is employed

(Fig. 6.7b), and when a RANDOM heuristic – a random walk of the search space – is employed (Fig. 6.7c). In all cases, the trajectory starts from the lower left corner of the map. For easier visualization, the paths are displayed on a grayscaled image of the search map in Fig. 6.4 with the path color changing from blue to yellow for visual tracking of the paths. The trajectory in Fig. 6.7a mostly visits the higher-reward cells and yields near-optimal performance. The vehicle makes fewer turns compared to Fig. 6.7b and Fig. 6.7c. We note that while the trajectory in Fig. 6.7c successfully avoids visiting lower-reward cells, it is not ideal for the search applications that we are interested in in this dissertation due to the large number of turns the vehicle makes. The trajectory in Fig. 4b attempts to strike a balance between the two approaches by attempting to cover mostly high-reward cells while minimizing the number of turns performed.

Chapter 7

Conclusions and Future Work

In this dissertation, we address the problem of computing efficient search plans for robotic applications. We develop decision-theoretic cost functions to assess the value of sampling from a search location, and we propose novel approaches to compute near-optimal search plans in feasible time. We are inspired by subsea search applications such as mine-hunting missions and search and rescue operations. However, our results inform a wide-range of real-world robotic applications. Our decision-theoretic cost function to compute the optimal search locations account for multiple targets, false detections and environmental uncertainty, which are often ignored in the search literature. In order to address the computational challenge of computing near-optimal paths in real-time, we employ an exact and an approximate branch-and-bound planner and a Monte Carlo tree search planner. The results of the numerical illustrations show that our proposed approaches yield improved search performance.

When environmental information can be acquired to improve the performance of a search mission, we consider different scenarios where the search sensor and the environmental characterization sensor can be placed on the same vehicle or on separate vehicles. For each scenario, we derive a decision-theoretic cost function to compute the locations where environmental information should be acquired. We show that when the search sensor and the environmental characterization sensor are placed on separate vehicles, environmental information should be acquired at the locations where the greatest reduction of the uncertainty in anticipated estimation accuracy will occur. For the case where the search sensor and the environmental characterization sensor are placed on the same vehicle and both sensors op-

erate simultaneously, we show that the expected estimation accuracy should be maximized. Finally, for the case where the search sensor and the environmental characterization sensor are placed on the same vehicle but only one sensor can be active at a time, we derive a utility function that yields when and where to search and when and where to characterize the environment so that the probability of attaining a desired level of risk reduction is maximized. We show that if environmental characterization of a location is beneficial for follow-on search, then environmental characterization should be conducted as soon as possible during a mission so that in case the mission goals cannot be met under the present environmental conditions, the sensing agent will be freed up sooner.

When there are multiple vehicles to search the environment and the goal is to maximize a joint search performance, we consider the case the vehicles can communicate with each other to share their local observations, but communication is limited and it incurs a cost. Thus, each vehicle has to reason about the cost and the value of communicating with another vehicle. Our approach to reduce the computational complexity of the problem is to separate path planning from communication planning and execute path planning in plan-time and communication planning in run-time. We also show an efficient method to reduce the number of computations each vehicle performs to obtain tractable solutions. The results of our preliminary results show promise to tackle the multi-vehicle search problem.

Our current utility function to assess the value of searching a location does not account for spatially correlated target density. Thus, a possible extension of our work is to consider spatial correlation among neighboring cells. Another possible extension of this work is to address the cases where provable near-optimal search paths can be computed in polynomial time. In addition, our approach to multi-vehicle search problem can be generalized to a general mapping problem where a number of agents are tasked with collaboratively mapping the environment.

Bibliography

[1] J. M. Cozzolino, "Sequential search for an unknown number of objects of nonuniform size," *Operations Research*, vol. 20, no. 2, pp. 293–308, 1972.

[2] G. Davis, "Mineral exploration decisions: a guide to economic analysis and modeling," *Mineralogical Magazine*, vol. 55, no. 380, pp. 488–489, 1991.

[3] I. Abi-Zeid and J. R. Frost, "SARPlan: A decision support system for Canadian search and rescue operations," *European Journal of Operational Research*, vol. 162, no. 3, pp. 630–653, 2005.

[4] T. M. Kratzke, L. D. Stone, and J. R. Frost, "Search and rescue optimal planning system," in *13th Conference on Information Fusion*. IEEE, 2010, pp. 1–8.

[5] D. Cooper, J. Frost, and R. Q. Robe, "Compatibility of land SAR procedures with search theory," Technical Report, 2003.

[6] G. A. Gorry, J. P. Kassirer, A. Essig, and W. B. Schwartz, "Decision analysis as the basis for computer-aided management of acute renal failure," *The American Journal of Medicine*, vol. 55, no. 4, pp. 473–484, 1973.

[7] M. Mangel and C. W. Clark, "Uncertainty, search, and information in fisheries," *ICES Journal of Marine Science*, vol. 41, no. 1, pp. 93–103, 1983.

[8] M. Mangel and J. H. Beder, "Search and stock depletion: theory and applications," *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 42, no. 1, pp. 150–163, 1985.

[9] S. Bryant, "Advanced unmanned search system (AUSS) performance analysis," Naval Ocean Systems Center, San Diego, CA, Tech. Rep., 1979.

[10] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, 2005.

[11] B. O. Koopman, "The theory of search: III. The optimum distribution of searching effort," *Operations Research*, vol. 5, no. 5, pp. 613–626, 1957.

[12] J. De Guenin, "Optimum distribution of effort: an extension of the Koopman basic theory," *Operations Research*, vol. 9, no. 1, pp. 1–7, 1961.

[13] D. F. Mela, "Letter to the editor: Information theory and search theory as special cases of decision theory," *Operations Research*, vol. 9, no. 6, pp. 907–909, 1961.

[14] J. B. Kadane, "Optimal whereabouts search," *Operations Research*, vol. 19, no. 4, pp. 894–904, 1971.

[15] M. C. Chew Jr, "Optimal stopping in a discrete search problem," *Operations Research*, vol. 21, no. 3, pp. 741–747, 1973.

[16] S. M. Pollock, "Sequential search and detection," Massachuestts Institute of Technology, Cambridge Operations Research Center, Tech. Rep., 1964.

[17] L. D. Stone, J. A. Stanshine, and C. A. Persinger, "Optimal search in the presence of Poisson-distributed false targets," *SIAM Journal on Applied Mathematics*, vol. 23, no. 1, pp. 6–27, 1972.

[18] J. M. Dobbie, "Some search problems with false contacts," *Operations Research*, vol. 21, no. 4, pp. 907–925, 1973.

[19] D. Kalbaugh, "Optimal search among false contacts," *SIAM Journal on Applied Mathematics*, vol. 52, no. 6, pp. 1722–1750, 1992.

[20] M. Kress, K. Y. Lin, and R. Szechtman, "Optimal discrete search with imperfect specificity," *Mathematical Methods of Operations Research*, vol. 68, no. 3, pp. 539–549, 2008.

[21] T. H. Chung and J. W. Burdick, "Analysis of search decision making using probabilistic search strategies," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 132–144, 2012.

[22] H. Luss, "Multiperiod search models for an unknown number of valuable objects," *Decision Sciences*, vol. 6, no. 3, pp. 430–438, 1975.

[23] F. H. Smith and G. Kimeldorf, "Discrete sequential search for one of many objects," *The Annals of Statistics*, vol. 3, no. 4, pp. 906–915, 1975.

[24] G. Kimeldorf and F. H. Smith, "Binomial searching for a random number of multinomially hidden objects," *Management Science*, vol. 25, no. 11, pp. 1115–1126, 1979.

[25] E.-M. Wong, F. Bourgault, and T. Furukawa, "Multi-vehicle Bayesian search for multiple lost targets," in *Proceedings of the International Conference on Robotics and Automation*. IEEE, 2005, pp. 3169–3174.

[26] H. Lau, "Optimal search in structured environments," Ph.D. dissertation, University of Technology, Sydney, 2007.

[27] J. B. Kadane, "Discrete search and the Neyman-Pearson lemma," *Journal of Mathematical Analysis and Applications*, vol. 22, no. 1, pp. 156–171, 1968.

[28] I. Wegener, "The discrete search problem and the construction of optimal allocations," *Naval Research Logistics (NRL)*, vol. 29, no. 2, pp. 203–212, 1982.

[29] H. Sato and J. O. Royset, "Path optimization for the resource-constrained searcher," *Naval Research Logistics (NRL)*, vol. 57, no. 5, pp. 422–440, 2010.

[30] S. P. Kragelund, "Optimal sensor-based motion planning for autonomous vehicle teams," Ph.D. dissertation, Monterey, California: Naval Postgraduate School, 2017.

[31] I. Wegener, "Optimal search with positive switch cost is NP-hard," *Information Processing Letters*, vol. 21, no. 1, pp. 49–52, 1985.

[32] H. Richardson, "ASW information processing and optimal surveillance in a false target environment," Office of Naval Research Center, Arlington, VA, Tech. Rep., 1973.

[33] W. Barker and D. Wagner, "Information theory and optimal detection search," in *SIAM Reviews*, vol. 18, no. 4, 1976.

[34] Z. A. Saigol, "Automated planning for hydrothermal vent prospecting using AUVs," Ph.D. dissertation, University of Birmingham, 2011.

[35] M. Harris, W. Avera, C. Steed, J. Sample, L. D. Bibee, D. Morgerson, J. Hammack, and M. Null, "AQS-20 through-the-sensor (TTS) performance assessment," in *Proc. IEEE/MTS OCEANS*, Washington, DC, USA, 2005, pp. 460–465.

[36] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman, "Transition-independent decentralized Markov decision processes," in *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*. ACM, 2003, pp. 41–48.

[37] B. Eker, E. Özkucur, C. Meriçli, T. Meriçli, and H. L. Akin, "A finite horizon DEC-POMDP approach to multi-robot task learning," in *5th International Conference on Application of Information and Communication Technologies (AICT)*. IEEE, 2011, pp. 1–5.

[38] J. G. Baylog and T. A. Wettergren, "Multiple pass collaborative search in the presence of false alarms," in *SPIE Defense+ Security*, April, 2015.

[39] V. Myers and D. P. Williams, "A POMDP for multi-view target classification with an autonomous underwater vehicle," in *Proc. IEEE/MTS OCEANS*, Seattle, WA, USA, 2010.

[40] Y. Zhang, A. B. Baggeroer, and J. G. Bellingham, "Spectral-feature classification of oceanographic processes using an autonomous underwater vehicle," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 726–741, 2001.

[41] B. H. Houston, J. A. Bucaro, T. Yoder, L. Kraus, J. Tressler, J. Fernandez, T. Montgomery, and T. Howarth, "Broadband low frequency sonar for non-imaging based identification," in *Proc. IEEE/MTS OCEANS*, Biloxi, MS, USA, 2002, pp. 383–387.

[42] M. P. Hayes and P. T. Gough, "Synthetic aperture sonar: A review of current status," *IEEE Journal of Oceanic Engineering*, vol. 34, no. 3, pp. 207–224, July 2009.

[43] M. C. Coleman and D. E. Block, "Nonlinear experimental design using Bayesian regularized neural networks," *AIChE Journal*, vol. 53, no. 6, pp. 1496–1509, 2007.

[44] C. Papadimitriou, J. L. Beck, and S.-K. Au, "Entropy-based optimal sensor location for structural model updating," *Journal of Vibration and Control*, vol. 6, no. 5, pp. 781–800, 2000.

[45] A. Elfes, "Dynamic control of robot perception using multi-property inference grids," in *Proceedings of the International Conference on Robotics and Automation.* IEEE, 1992, pp. 2561–2567.

[46] P. Elmore, W. E. Avera, M. M. Harris, and K. M. Duvieilh, "Environmental measurements derived from tactical mine-hunting sonar data," in *Proc. IEEE/MTS OCEANS*, 2007.

[47] A. Zare and J. T. Cobb, "Sand ripple characterization using an extended synthetic aperture sonar model and MCMC sampling methods," in *IEEE/MTS OCEANS*, San Diego, CA, USA, 2013, pp. 1–7.

[48] K. Takahashi, J. Igel, and H. Preetz, "Clutter modeling for ground-penetrating radar measurements in heterogeneous soils," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, no. 4, pp. 739–747, 2011.

[49] K. Takahashi, H. Preetz, and J. Igel, "Soil properties and performance of landmine detection by metal detector and ground-penetrating radar—soil characterisation and its verification by a field test," *Journal of Applied Geophysics*, vol. 73, no. 4, pp. 368–377, 2011.

[50] P. D. Gader, M. Mystkowski, and Y. Zhao, "Landmine detection with ground penetrating radar using hidden Markov models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 6, pp. 1231–1244, 2001.

[51] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings of International Symposium on Computational Intelligence in Robotics and Automation.* IEEE, July 1997, pp. 146–151.

[52] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *Proceedings of the International Conference on Intelligent Robots and Systems*. IEEE, 2002, pp. 540–545.

[53] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An experiment in integrated exploration," in *Proceedings of the International Conference on Intelligent Robots and Systems*. IEEE, 2002, pp. 534–539.

[54] H. Carrillo, P. Dames, V. Kumar, and J. A. Castellanos, "Autonomous robotic exploration using occupancy grid maps and graph SLAM based on Shannon and Rényi Entropy," in *Proceedings of the International Conference on Robotics and Automation*. IEEE, May 2015, pp. 487–494.

[55] S. Jaramillo and G. Pawlak, "AUV-based bed roughness mapping over a tropical reef," *Coral Reefs*, vol. 30, no. 1, pp. 11–23, 2011.

[56] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.

[57] T. R. Clem, "Sensor technologies for hunting buried sea mines," in *IEEE/MTS OCEANS*, vol. 1, 2002, pp. 452–460.

[58] J. A. Bucaro, Z. J. Waters, B. H. Houston, H. J. Simpson, A. Sarkissian, S. Dey, and T. J. Yoder, "Acoustic identification of buried underwater unexploded ordnance using a numerically trained classifier (L)," *The Journal of the Acoustical Society of America*, vol. 132, no. 6, pp. 3614–3617, 2012.

[59] M. P. Hayes and P. T. Gough, "Broad-band synthetic aperture sonar," *IEEE Journal of Oceanic Engineering*, vol. 17, no. 1, pp. 80–94, Jan 1992.

[60] J. C. Gittins, "Bandit processes and dynamic allocation indices," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 148–177, 1979.

[61] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[62] J.-Y. Audibert, R. Munos, and C. Szepesvári, "Exploration–exploitation tradeoff using variance estimates in multi-armed bandits," *Theoretical Computer Science*, vol. 410, no. 19, pp. 1876–1902, 2009.

[63] K. K. Damghani, M. Taghavifard, and R. T. Moghaddam, "Decision making under uncertain and risky situations," in *Enterprise Risk Management Symposium Monograph Society of Actuaries-Schaumburg, Illinois*, vol. 15, 2009.

[64] J. Bowen and Z.-l. Qiu, "Satisficing when buying information," *Organizational Behavior and Human Decision Processes*, vol. 51, no. 3, pp. 471–481, 1992.

[65] M. I. Henig, "Risk criteria in a stochastic knapsack problem," *Operations Research*, vol. 38, no. 5, pp. 820–825, 1990.

[66] K. E. Wilson, R. Szechtman, and M. P. Atkinson, "A sequential perspective on searching for static targets," *European Journal of Operational Research*, vol. 215, no. 1, pp. 218–226, 2011.

[67] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of Markov decision processes," *Mathematics of Operations Research*, vol. 12, no. 3, pp. 441–450, 1987.

[68] J. Pineau, "Tractable planning under uncertainty: exploiting structure," Ph.D. dissertation, Carnegie Mellon University, the Robotics Institute, 2004.

[69] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces." in *Robotics: Science and Systems*, vol. 2008. Zurich, Switzerland., 2008.

[70] R. He, E. Brunskill, and N. Roy, "PUMA: Planning under uncertainty with macro-actions." in *AAAI*, 2010.

[71] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," in *Advances in Neural Information Processing Systems*, 2010, pp. 2164–2172.

[72] N. Ye, A. Somani, D. Hsu, and W. S. Lee, "DESPOT: Online POMDP planning with regularization," *Journal of Artificial Intelligence Research*, vol. 58, pp. 231–266, 2017.

[73] L. Peshkin, K.-E. Kim, N. Meuleau, and L. P. Kaelbling, "Learning to cooperate via policy search," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, 2000, pp. 489–496.

[74] D. V. Pynadath and M. Tambe, "The communicative multiagent team decision problem: Analyzing teamwork theories and models," *Journal of Artificial Intelligence Research*, vol. 16, pp. 389–423, 2002.

[75] P. J. Gmytrasiewicz and P. Doshi, "A framework for sequential planning in multi-agent settings," *Journal of Artificial Intelligence Research*, vol. 24, pp. 49–79, 2005.

[76] D. S. Bernstein, S. Zilberstein, and N. Immerman, "The complexity of decentralized control of Markov decision processes," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, 2000, pp. 32–37.

[77] J. S. Dibangoye, C. Amato, A. Doniec, and F. Charpillet, "Producing efficient error-bounded solutions for transition independent decentralized MDPs," in *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*, 2013, pp. 539–546.

[78] P. Velagapudi, P. Varakantham, K. Sycara, and P. Scerri, "Distributed model shaping for scaling to decentralized POMDPs with hundreds of agents," in *The 10th International Conference on Autonomous Agents and Multi-Agent Systems*, vol. 3, 2011, pp. 955–962.

[79] C. Amato, G. Konidaris, G. Cruz, C. A. Maynor, J. P. How, and L. P. Kaelbling, "Planning for decentralized control of multiple robots under uncertainty," in *Proceedings of the International Conference on Robotics and Automation*. IEEE, 2015, pp. 1241–1248.

[80] C. V. Goldman and S. Zilberstein, "Optimizing information exchange in cooperative multi-agent systems," in *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2003, pp. 137–144.

[81] R. Nair, M. Roth, and M. Yohoo, "Communication for improving policy computation in distributed POMDPs," in *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, vol. 3. IEEE, 2004, pp. 1098–1105.

[82] S. A. Williamson, E. H. Gerding, and N. R. Jennings, "A principled information valuation for communications during multi-agent coordination," in *Proc. of AAMAS Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains*, 2008, pp. 137–151.

[83] ——, "Reward shaping for valuing communications during multi-agent coordination," in *Proceedings of The 8th International Conference on Autonomous Agents and Multi-Agent Systems*, vol. 1, 2009, pp. 641–648.

[84] R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun, "Game theoretic control for robot teams," in *Proceedings of the International Conference on Robotics and Automation*. IEEE, 2005, pp. 1163–1169.

[85] C. V. Goldman and S. Zilberstein, "Decentralized control of cooperative systems: Categorization and complexity analysis," *Journal of Artificial Intelligence Research*, vol. 22, pp. 143–174, 2004.

[86] R. Becker, A. Carlin, V. Lesser, and S. Zilberstein, "Analyzing myopic approaches for multi-agent communication," *Computational Intelligence*, vol. 25, no. 1, pp. 31–50, 2009.

[87] A. Carlin and S. Zilberstein, "Myopic and non-myopic communication under partial observability," in *Proceedings of the International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, vol. 2. IEEE, 2009, pp. 331–338.

[88] P. Xuan, V. Lesser, and S. Zilberstein, "Communication decisions in multi-agent cooperation: Model and experiments," in *Proceedings of the 5th International Conference on Autonomous Agents*, 2001, pp. 616–623.

[89] F. Wu, S. Zilberstein, and X. Chen, "Online planning for multi-agent systems with bounded communication," *Artificial Intelligence*, vol. 175, no. 2, pp. 487–511, 2011.

[90] M. Roth, R. Simmons, and M. Veloso, "Reasoning about joint beliefs for execution-time communication decisions," in *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2005, pp. 786–793.

[91] M. Roth, "Execution-time communication decisions for coordination of multi-agent teams," Ph.D. dissertation, Carnegie Mellon University, the Robotics Institute, 2007.

[92] D. S. Bernstein, E. A. Hansen, and S. Zilberstein, "Bounded policy iteration for decentralized POMDPs," in *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 2005, pp. 52–57.

[93] V. V. Unhelkar and J. A. Shah, "ConTaCT: Deciding to communicate during time-critical collaborative tasks in unknown, deterministic domains." in *AAAI*, 2016, pp. 2544–2550.

[94] D. Szer and F. Charpillet, "An optimal best-first search algorithm for solving infinite horizon DEC-POMDPs," in *European Conference on Machine Learning*. Springer, 2005, pp. 389–399.

[95] F. A. Oliehoek, S. Whiteson, and M. T. Spaan, "Lossless clustering of histories in decentralized POMDPs," in *Proceedings of The 8th International Conference on Autonomous Agents and Multi-Agent Systems*, vol. 1, 2009, pp. 577–584.

[96] E. A. Hansen, D. S. Bernstein, and S. Zilberstein, "Dynamic programming for partially observable stochastic games," in *AAAI*, vol. 4, 2004, pp. 709–715.

[97] D. Szer and F. Charpillet, "Point-based dynamic programming for DEC-POMDPs," in *AAAI*, vol. 6, 2006, pp. 1233–1238.

[98] D. S. Bernstein, C. Amato, E. A. Hansen, and S. Zilberstein, "Policy iteration for decentralized control of Markov decision processes," *Journal of Artificial Intelligence Research*, vol. 34, no. 1, p. 89, 2009.

[99] S. Seuken and S. Zilberstein, "Memory-bounded dynamic programming for DEC-POMDPs." in *IJCAI*, 2007, pp. 2009–2015.

[100] C. Amato, J. S. Dibangoye, and S. Zilberstein, "Incremental policy generation for finite-horizon DEC-POMDPs." in *ICAPS*, 2009.

[101] A. Singh, A. Krause, C. Guestrin, W. J. Kaiser, and M. A. Batalin, "Efficient planning of informative paths for multiple robots." in *IJCAI*, vol. 7, 2007, pp. 2204–2211.

[102] G. Hollinger, S. Singh, J. Djugash, and A. Kehagias, "Efficient multi-robot search for a moving target," *The International Journal of Robotics Research*, vol. 28, no. 2, pp. 201–219, 2009.

[103] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica: Journal of the Econometric Society*, pp. 497–520, 1960.

[104] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations Research*, vol. 14, no. 4, pp. 699–719, 1966.

[105] B. W. Wah and C. F. Yu, "Stochastic modeling of branch-and-bound algorithms with best-first search," *IEEE Transactions on Software Engineering*, vol. SE-11, no. 9, pp. 922–934, Sept 1985.

[106] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, and E. C. Sewell, "Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning," *Discrete Optimization*, vol. 19, pp. 79–102, 2016.

[107] S. Gelly, L. Kocsis, M. Schoenauer, M. Sebag, D. Silver, C. Szepesvári, and O. Teytaud, "The grand challenge of computer Go: Monte Carlo tree search and extensions," *Communications of the ACM*, vol. 55, no. 3, pp. 106–113, 2012.

[108] M. H. Winands, Y. Bjornsson, and J.-T. Saito, "Monte Carlo tree search in lines of action," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 4, pp. 239–250, 2010.

[109] A. Arora, R. Fitch, and S. Sukkarieh, "An approach to autonomous science by modeling geological knowledge in a Bayesian framework," in *Proceedings of the International Conference on Intelligent Robots and Systems*. IEEE, Sept 2017, pp. 3803–3810.

[110] J. L. Nguyen, N. R. J. Lawrance, R. Fitch, and S. Sukkarieh, "Real-time path planning for long-term information gathering with an aerial glider," *Autonomous Robots*, vol. 40, no. 6, pp. 1017–1039, Aug 2016.

[111] L. Kocsis and C. Szepesvári, "Bandit based Monte Carlo planning," in *European Conference on Machine Learning*. Springer, 2006, pp. 282–293.

[112] A. Rimmel, F. Teytaud, and T. Cazenave, "Optimization of the nested Monte Carlo algorithm on the traveling salesman problem with time windows," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2011, pp. 501–510.

[113] T. J. Walsh, S. Goschin, and M. L. Littman, "Integrating sample-based planning and model-based reinforcement learning." in *AAAI*, 2010.

[114] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of Monte Carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012.

[115] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[116] J. McMahon, H. Yetkin, A. Wolek, Z. Waters, and D. J. Stilwell, "Towards real-time search planning in subsea environments," in *International Conference on Intelligent Robots and Systems*, Vancouver, BC, Canada, 2017.

[117] D. Feillet, P. Dejax, and M. Gendreau, "Traveling salesman problems with profits," *Transportation Science*, vol. 39, no. 2, pp. 188–205, 2005.

[118] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem: A survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, 2011.

[119] G. Laporte and S. Martello, "The selective traveling salesman problem," *Discrete Applied Mathematics*, vol. 26, no. 2-3, pp. 193–207, 1990.

[120] S. Martello and P. Toth, "Algorithms for knapsack problems," *North-Holland Mathematics Studies*, vol. 132, pp. 213–257, 1987.

[121] W. Zhang, "Depth-first branch-and-bound versus local search: A case study," in *National Conference on Artificial Intelligence*. Austin, TX, USA: AAAI, 2000, pp. 930–935.

[122] G. A. Croes, "A method for solving traveling-salesman problems," *Operations Research*, vol. 6, no. 6, pp. 791–812, 1958.

[123] C. Sanderson, "Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments," 2010.

[124] G. Tang, Z. Wang, and A. Williams, "On the construction of an optimal feedback control law for the shortest path problem for the Dubins car-like robot," in *Proceedings of the 30th Southeastern Symposium on System Theory*. IEEE, 1998, pp. 280–284.

[125] H. Yetkin, J. McMahon, N. Topin, A. Wolek, Z. Waters, and D. J. Stilwell, "Online planning for unmanned vehicles performing information gathering tasks in large state spaces," in *In preparation for submission*.

[126] O. Tange, "GNU parallel - the command-line power tool," *The USENIX Magazine*, vol. 36, no. 1, pp. 42–47, 2011.

[127] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*. Springer Science & Business Media, 2013.

Appendices

Appendix A

Bayes Estimators for Specific Value Functions

Let $\theta \in \Theta$ be such that $\theta_1 \leq \theta_2 \leq \ldots \leq \theta_n$. After acquiring a measurement h, we form an estimate $\delta(h)$ on θ. We now define two decision-theoretic functions, the zero-one utility function and the linear loss function, to evaluate the value of forming the estimate $\delta(h)$ on θ.

Zero-One Utility Function

Given the measured data h, a zero-one utility function defines the utility of the estimate $\delta(h)$ when θ is the true value.

$$U\big(\theta, \delta(h)\big) = \begin{cases} 1 & \text{if } \theta = \delta(h) \\ 0 & \text{if } \theta \neq \delta(h) \end{cases} \tag{A.1}$$

Linear Loss Function

Given the measured data h, a linear loss function defines the loss of the estimate $\delta(h)$ when θ is the true value.

$$L_X\big(\theta, \delta(h)\big) = \begin{cases} c_1\big(\theta - \delta(h)\big) & \delta(h) < \theta \\ c_2\big(\delta(z) - \theta\big) & \delta(h) \geq \theta \end{cases} \tag{A.2}$$

where $c_1 > 0$ and $c_2 > 0$ are relative costs of overestimation ($\delta(h) > \theta$) and underestimation ($\delta(h) < \theta$).

In decision theory, a common approach to choose a decision rule is to compute the Bayes estimator which minimizes the posterior expected value of a loss function, or equivalently, maximizes the posterior expected value of a utility function. The following propositions specify the corresponding Bayes estimates for the zero-one utility function in (A.1) and the linear-loss function in (A.2).

Proposition 1. The Bayes estimate with respect to the zero-one utility function in (A.1) is the mode of the belief distribution $P\big(\theta \mid h\big)$

$$\delta^\star(h) = \max_\theta P\big(\theta \mid h\big) \tag{A.3}$$

Proposition 2. Let $F_{\theta|h}(a)$ be such that

$$\begin{aligned} F_{\theta|h}(a) &:= P\big(\theta \leq a \mid h\big) \\ &= \sum_{i=0}^{a} P\big(\theta = i \mid h\big) \end{aligned} \tag{A.4}$$

Then, the Bayes estimate with respect to the linear-loss function in (A.2) is

$$\delta^\star(h) = \theta_{l+1} \tag{A.5}$$

where

$$l = \arg\max_a \left(F_{\theta|h}(a) \leq \frac{c_1}{c_1 + c_2}\right) \tag{A.6}$$

Proofs for these propositions can be found in any standard book on statistical decision (see, for example, [127]).