Optimization Problems in Sensor Network Data Collection

by

Simon Shamoun

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2011

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirements for the degree of Doctor of Philosophy.

Amotz Bar-Noy

Date | Chair of Examining Committee

Theodore Brown

Date | Executive Officer

Amotz Bar-Noy

Theodore Brown

Ping Ji

David Sarne

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

ABSTRACT

OPTIMIZATION PROBLEMS IN SENSOR NETWORK DATA COLLECTION

BY

SIMON SHAMOUN

Advisor: Amotz Bar-Noy

Data collection is one of the most important tasks of many sensor networks. The data collected by sensors is used to monitor and analyze various systems, such as volcanoes, forests, and bridges. Large scale wireless sensor networks can provide timely access to a wealth of data, but obtaining this data is challenged by various resource constraints. This thesis proposes and analyzes solutions to three optimization problems that arise from the conflict between data collection and resource constraints: (1) maximize coverage by a set of sensors when the coverage they provide varies with location; (2) select a subset of the sensors, within some budget constraint, that best predict the data streams produced by all the sensors in the network; and (3) minimize the cost needed to find the top ranking sensor readings according to some criteria. The analyses of these problems use three different views of a sensor network: a coverage-centric view, in which each sensor is valued for its coverage ability; a data-centric view, in which each sensor is valued for the data it provides; and an agent-centric view, in which each sensor

is viewed as an independent agent with information of value to the application. By choosing an appropriate view of the network, it is possible to separate the analysis from implementation details and apply well-established techniques from other domains to the problem solution. In this case, methodologies from stochastic and computational geometry, graph theory, and search theory are applied to the respective problems. This thesis presents optimal solutions to the coverage and search problems, approximation bounds on the best possible solution to the selection problem, and quantitative comparisons to alternative solutions to each problem in synthetic environments.

Acknowledgments

When I was about fifteen years old, my brother gave me a comic book called "School is Hell" by Matt Groening. I always remembered one cartoon of an old man, shriveled like a raisin, pressing a horn to his ear and saying "Speak up, sonny, I can't hear you" when he was asked, "How does it feel to finally finish your dissertation?" With help from above, I have completed my doctoral dissertation. As it turns out, I have been using my brother's old cell phone. I have trouble hearing on it, so I am often yelling at people to speak up.

First of all, I am very grateful to the late Thomas J. Sweeney and his spouse. Their generous contribution to New York University allowed my to get my bachelors degree without the worry of tuition payments. From this, I was able to achieve my remaining accomplishments.

Many people offered me guidance over the years that brought me to this point. Some of them are Professors Dennis Shasha from NYU, John Kender from Columbia University, Robert Goldberg from Queens College, and of course Ted Brown and Amotz Bar-Noy from the Graduate Center. Since I received my bachelors degree, Dennis has regularly been a source of guidance and referred me to various people about jobs and help with my education. Professor Kender explained to me well the reasons why I would choose to get a Ph.D., and Dr. Goldberg gave me some simple but good advice on how to proceed. I would also like to thank my first research advisor, Neng-Fa Zhou.

My teaching experience at Queens College finally convinced me to pursue the

Ph.D. This was made possible by the department chairs, Tsaiyun Ihsin Phillips, Jenny Whitehead, Zhigang Xiang, who gave me the job; Sandy Cribbs, Xiu-Yi Huang, and Keisha Williams, who always helped me with a smile; and the other staff and faculty members that assisted me. I want to especially thank Dr. Jerry Waxman, who I regularly turned to for guidance, and Dr. Bojana Obrenic. Because of her rigor and guidance in teaching discrete structures, I acquired the skills I needed to work in theoretical computer science.

My family has been supportive throughout. Thanks to my mother, Rina, for your persistence and my sister Judy, my brother Alan, and your families for your support. Our father's pride in our accomplishments helped us all in our success.

I feel like we were a family at the Graduate Center. I don't want to name every single person in the department, but the camaraderie of everybody in it counts. Thanks to Joe Driscoll, Lina Garcia, and Kara Heffernan for all your help with administrative matters; Ali Assarpour, Panos Cheilaris, Yi Feng, and Matt Johnson for your insights, contributions, and assistance; Amotz and Stathis Zachos, for giving us the tools we need to do our work; and Ping Ji, for serving on my committee.

Sarit Kraus made my work with David Sarne at Bar Ilan University possible. She is also an invaluable advisor to David.

Finally, I want to thank the three people I worked with most closely in the last four years. They not only guided me in research and career decisions, but they tried their best to serve as friends, lifetime coaches, and sources of entertainment. Thank you to Ted Brown, David Sarne, and especially Amotz.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Data collection is one of the most important tasks of many sensor networks. The data collected by sensors is used to monitor and analyze various systems, such as amphibian populations [25, 53], volcanoes [62, 63], and bridges [28]. Large scale wireless sensor networks can greatly enhance the study of these systems by providing timely access to a wealth of data, but obtaining this data is challenged by various resource constraints. The following two examples highlight some of the benefits and challenges encountered in deploying sensor networks:

Amphibian monitoring. The nearly uninhibited spread of cane toads in north-eastern Australia may negatively impact the local environment [25]. Scientists would like to monitor their population and movement to understand their impact on the environment and possibly control their spread. This can be done using large scale wireless networks of acoustic sensors. An early system used expensive stand-alone devices that required expensive, infrequent, on-site data collec-

tion. The use of low cost sensors equipped with wireless communications devices enables unmonitored coverage of large areas with quick feedback. The acoustic signals are used to recognize and classify frog vocalizations. Since only the presence of cane toads is needed, recognition and classification can be performed in-network by resource-rich but costly intermediary nodes that merely report the presence of cane toads to a central base station. The low-cost motes do not have the processing power to perform these tasks, and it is too costly to transmit all acoustic signals to an off-line server for processing. Even with this hybrid approach, bandwidth limitations prevent continuous streaming of data by any individual mote, and sampling scheduling is required to avoid collisions and extend network lifetime.

Volcano monitoring. Scientists collect data about volcanic activity to monitor hazards and to understand the physical processes that occur within a volcano [62, 63]. Typical volcanic data-collection station are too large and heavy to deploy more than a few of them in remote or hazardous areas. Additionally, significant effort is required to manually retrieve data from them every few weeks. Better study of volcanic processes is possible by deploying large wireless networks of small, lightweight sensors. In one study, a network of sixteen motes monitored a volcano in Ecuador. It was impossible to continuously transmit all sensor data because of the high data rates and low bandwidth involved. Instead, nodes reported interesting seismic events that were detected locally to the base station. If the base station received enough event reports in a short time interval, it attempted to download the last sixty seconds of data from each node. Although complete data

is required to understand long-term trends, much of the data analysis focuses on discrete events.

In these examples, large scale sensor networks enhance the study of the respective systems by monitoring large areas for the presence of cane toads and collecting diverse signals from volcanic activity. The deployment of these networks is enabled by the use of small, low-cost wireless sensors, due to the low cost of acquiring sensor nodes, the ease deploying them, and the ease of collecting data from them. The lifetimes of these networks are constrained by the limited power supply to the nodes, requiring either duty cycling coupled with redundant coverage or manual visits to replace the power supplies. The greater challenge is in overcoming bandwidth limitations. In the case of cane toad monitoring, this a consequence of the limited computing power of the sensor devices, since they have to offload processing to an intermediate device; transferring all data is an actual requirement of volcano monitoring. This challenge is met by limiting the data collected or, in the case of cane toad monitoring, spending more money to increase the number of intermediate nodes.

The conflict between data collection and resource limitations results in various optimization problems. In an optimization problem, the goal is either to maximize some objective function given a restriction on the available resources, or to minimize the resources required to meet some objective function value. For example, the objective in cane toad monitoring is to maximize the area covered over a period of time given restrictions on the number of devices, the available power supply, and the available bandwidth. The objective in volcano monitoring is to

maximize the data gathered given limitations on bandwidth. This thesis analyzes three specific optimization problems in sensor networks:

- **Zone allocation.** Sensor coverage depends on surrounding conditions such as terrain, weather, and obstacles. Given a field of interest divided into zones of homogeneous conditions, such that the sensing area of a sensor is the same at all points in a zone, find the allocation of a set of sensors to the zones that maximizes the total area covered.

- **Sensor selection.** The number of continuous data streams produced by a large sensor network is a challenge from a variety of cost-driven perspectives, such as power supply and transmission, storage, and processing capacities. When there are predictability relationships between sensor streams, it is possible to collect data from only a subset of the streams and predict the values of the remaining streams. Given a cost assignment to all sensors and a budget constraint, select the set of sensor that best predicts all sensor streams in the network.

- **Best-valued data search.** The application that uses the sensor data may only need the sensor readings that are of most value to the application. Requesting, obtaining, and processing sensor readings all incur a cost to both the individual sensors and host device of the application. Given the number of readings required by the application, obtain the readings of most value to the application at least possible cost to the entire network.

These problems reflect three alternative views of a sensor network: a coverage-centric, data-centric, and agent-centric view, respectively. In the coverage-centric view, each sensor is a device that covers some amount of area, and the objective is to maximize the area covered by a set of sensors. In the data-centric view, each sensor is an object that provides a stream of data, and the objective is to maximize the quality of information obtained from the sensors. In the agent-centric view, each sensor is an independent agent with information of value to the application, and the objective is to minimize the cost needed to find the agents with the most valuable information. These three views can result in qualitatively different solutions. For example, a network that covers the most area around a volcano may miss the crucial points for understanding wave propagation. Likewise, determining the hazard level of an eruption may require knowledge of the strongest signals, while understanding the processes inside a volcano may require the broadest overview of all the signals. The analyses of these problems use methodologies from different domains–stochastic geometry, graph theory, and search theory, respectively.

For zone allocation, the following problem is specifically addressed: One is given a field divided into zones such that all sensors in a zone provide the same coverage. One can decide how to allocate sensors to each zone, but within each zone, sensors are randomly distributed. The objective is to maximize the area covered by a given number of sensors by choosing the best allocation. Two additional problems are studied: 1) What is the minimum number of sensors required to guarantee some expected level of coverage, and 2) how much additional

area is covered by the optimal allocation from some baseline allocation strategy. These problems are relevant to a strategist that would like to 1) maximize coverage on a limited budget; 2) minimize the costs required to guarantee some level of coverage; and 3) evaluate the benefit of zone allocation over distributing sensors uniformly throughout the entire field.

Even though better coverage is possible by deterministic deployment, the solution to the random deployment problem has some interesting features. For both random and deterministic deployment, optimizing the sensor allocation is difficult when considering the coverage sensors provide of bordering zones. Even when sensors only cover the zones in which they are allocated, it is difficult to accurately calculate the expected coverage by a random deployment and to determine the optimal arrangement of sensors in the deterministic case. Rather, the entire analysis ignores the effect of borders on expected coverage and optimal arrangements. Accordingly, the optimal allocation in both cases can be derived with generic solutions whose runtime increases with the number of sensors and zones. However, in the case of random deployment, the optimal allocation can be derived in runtime asymptotically equal to the number of zones by using a simple approximation of expected coverage. Despite these simplifications, simulations show that the actual coverage by this allocation is close to or even better than coverage by the best allocation found experimentally. The main purpose of the analysis, therefore, is to demonstrate how a simplification of expected coverage can still be used to efficiently derive a near-optimal solution to the zone allocation problem. A second objective is to demonstrate that zone allocation can be an

effective deterministic strategy, even under various simplifying assumptions.

A link- and data-driven technique is applied to the sensor selection problem. Sensor streams can be used to predict the streams of other sensors because of predictability relationships between them. For example, two nearby sensors will hear some of the same frog calls, although with different levels of distortion and time lag. The sensors and the predictability relationships between them can be represented by data objects and logical links, respectively, in an information network [38]. When a sensor stream is not available, one or more of its linked sensors can be used to predict it. The best choice of sensors depends on the strength of the relationships between the sensors and the topology of the corresponding relationships. In this thesis, regression analysis is used to model one stream by another [51, 57, 67], and the strength of the relationships is based on the accuracy of the regression model. This link- and data-driven approach allows the derivation of a general approximation algorithm for the problem. Also, by only including links for which predictability relationships exist, the computation needed to select the best set of sensors can be drastically reduced.

The best-value search is motivated by a pull approach to data collection, in which the base station actively searches for the data it needs rather than passively receiving data from sensors. The base station queries the network for the data it needs, and the nodes containing the data reply. If the query is for the data of most value to the base station, then knowledge of all data values is necessary at some level. This thesis presents a specific method for finding best-valued data referred to as “increasing threshold search”. Increasing threshold search is a type of search

with iteratively expanding search extents, in which a search is repeated in rounds with increasing extents until the search goal is met. In this case, the base station requests data whose value is below some threshold from all nodes (when the best value is the lowest value). If it does not receive any replies, it repeats the query with a higher threshold. It repeats this process until either it receives the required number of values or it requests data from all sensors, at which point it selects the number of values it needs. The expectation is that trading repetitive search costs with reduced costs associated with the search extents will result in a lower overall search cost. Increasing threshold search is analyzed independently from network topology and network protocols; it is only assumed that the base station can publish thresholds to all nodes in the network.

The next sections describe the background and related work that will assist in understanding the problems and outline the contributions this thesis makes to each problem.

1.1 Background and Related Work

Coverage by sensor networks has been studied under various models of sensor coverage. The binary sensing model is the simplest and most widely analyzed model of coverage provided by a sensor. In this model, a sensor detects all events in its sensing area. Brass [5] gives upper bounds on the capabilities of random and deterministic strategies in bounded and unbounded regions. Liu and Towsley [41] analyze different measures of coverage provided by random deployment in large

unbounded regions. Both assume uniformly sized disk shaped sensing areas for all sensors and location assignment according to a Poisson point distribution in random deployments. Lazos and Poovendran [36] provide a more robust analysis of coverage by random deployment. Their analysis accounts for border effects in bounded regions and the size and shape of each sensor. They provide a limited analysis of coverage by non-uniform distributions. Koskinen [30] analyzes the probability of full coverage in the limit. None of these works considered coverage in the terrain dependent model. Lan et al. [34] investigate the problem of minimizing the sensing radius required to asymptotically achieve full coverage of the unit square by uniform and non-uniform random distributions. For non-uniform distributions, they propose partitioning the square such that the distribution is uniform in each partition and setting the radii in each partition accordingly. Other models of coverage include the probabilistic sensing model [15, 68], in which each point in a sensor's sensing area is assigned a detection probability, and the general sensing model [41, 65], which considers the aggregation of signals received by sensors. In both models, unlike in the binary model, increasing the number of sensors covering a point and their proximity to it improves coverage.

Several papers consider environmental effects on coverage in designing deployment strategies. Yang et al. [66] derive the sensing areas of chemical sensors by correlating models of gas dispersion to actual terrain and weather data, while remaining papers consider a generic notion of environmental effects on coverage. The standard solution is to model the field of interest and sensor coverage with a grid, and greedily assign sensors grid locations [15, 66, 68]. The runtime com-

plexity of these algorithms is $O(N^2m)$, in which N is the number of grid points in the field and m is the number of sensors deployed, which is very costly for fine grids. Another approach [59] is to place sensors along a standard grid and adjust the distances between grid points to improve coverage. Without proportionate distribution of sensors to different areas of the field, this results in large uncovered areas. A novel approach to assigning sensor locations represents the field using a gray-scale image, in which the intensity varies with the sensing range, and then employs dithering algorithms to determine sensor locations in the field [60]. All of these solutions are for deterministic coverage; no strategies for random coverage under the location dependent coverage model were studied.

The sensor selection and placement problem has been addressed independently by the stream mining and sensor placement communities. The problem of optimal sensor placement has been studied in [32, 33]. However, this work explicitly focuses on sensor placement in the context of spatial phenomenon. The work in [39] studies the problem of sensor selection, when the costs and benefits of placing a sensor at a given location have already been modeled externally. The problem of sensor selection has been studied in detail in [19, 31]. The work in [31] is similar to [39], in that it examines the problem of sensor selection when the benefits of picking particular sets of sensors can be externally modeled. None of these techniques discuss models for determining the optimal sensor sets in a data-driven manner. The work in [19] is somewhat data-driven, in that it uses the current data in conjunction with external utility functions for the selection process. Many of the aforementioned techniques are not data-driven and require external

feedback about sensor benefits.

Increasing threshold search is modeled after expanding ring search, which is used to find the shortest route between two nodes in an ad hoc network. In expanding ring search, one node forwards a route request to all nodes within some hop count itself, and repeats the request with a greater limit on the hop count until either the target is found or the entire network is flooded with the route request. Expanding ring search can also be used in sensor networks to find specific data, as it is used in peer-to-peer networks. However, this is not the same as finding the best-valued data, which is the goal of increasing threshold search. Chang and Liu [7] show how to derive the optimal expanding ring search sequence using dynamic programming when the probability distribution of the minimum hop count to the target is known *a priori*. They prove that when the probability distribution is not known *a priori*, a randomized strategy is optimal, derive the optimal strategy, and prove that it has a competitive ratio of e [8]. Baryshnikov et al. [3], prove that the California split rule strategy (see Chapter 4) is the optimal deterministic strategy when the probability distribution is unknown and that it has a competitive ratio of 4. Similar techniques include iterative deepening [29, 55] and iterative broadening [18] depth-first search, whose expected costs were studied as well. The results of all these studies are inapplicable to increasing threshold search because the cost models differ, as explained in Chapter 4.

1.2 Contributions

The zone allocation problem is analyzed in Chapter 2. Closed-form formulas are derived to determine the optimal allocation in runtime asymptotically equal to the number of zones, which is the best possible runtime for any solution; to calculate the minimum number of sensors required to achieve a specific level of coverage; and to bound the maximum increase in coverage over a strategy oblivious to differences in sensing areas. Results show that this bound is no greater than 13% for a field with two zones. A simulation study is conducted to validate the analysis. The study uses a slightly more realistic model of sensor coverage than the standard disk sensing model by calculating sensing ranges using a weighted distance function [14]. The study shows that coverage by the allocation derived analytically is close to coverage by the allocation derived experimentally. It also shows that zone allocation for deterministic deployment results in coverage comparable to coverage by the standard greedy assignment algorithm, but in significantly less time.

In Chapter 3, the sensor selection problem is proven to be NP-complete and shown to be closely related to the generalized maximum coverage problem [13] and the budgeted maximum coverage problem [27]. Consequently, it is established that no approximation factor better than $(e-1)/e$ can be guaranteed by any polynomial-time algorithm unless $P = NP$. However, an adaptation of the greedy algorithm for the generalized maximum coverage problem has an approximation guarantee of at least $\frac{e-1}{2 \cdot e-1}$. The speed and quality of solutions derived

by this algorithm are compared to an integer programming based solution and a sampling based solution on real and synthetic data sets. The results show that the greedy algorithm is close to optimal with a runtime that parallels the basic sampling algorithm. Additionally, limiting the allowable virtual links greatly speeds up execution with a small change in quality of the solution.

In Chapter 4, the optimal increasing threshold search strategy for a single agent is derived for the case when thresholds can be selected from a continuous range of values. The analysis reveals that the thresholds in the optimal search sequence are characterized by a common probabilistic property. This enables the proof that the optimal sequence of thresholds is a single or an infinite number of thresholds. These are important results: While the optimal can no longer be derived using dynamic programming, since it is potentially infinite, the common probabilistic property facilitates the extraction of a distribution-independent solution, which can then be mapped to the sequence of actual thresholds for specific distributions using a simple transformation. A similar method of analysis is applied to the case of a search for the K best-valued agents, with similar results. For the case when thresholds can only be selected from a discrete range of values, it is shown how to derive the optimal sequence using dynamic programming.

The properties and performance of the optimal and alternative strategies are demonstrated by evaluation in a synthetic environment. The results primarily highlight the tradeoff between the expected number of search rounds and the expected number of agents found by the search. An important observation is that the expected number of search rounds is below five for all single agent search sce-

narios evaluated. This supports the applicability of the optimal strategy: Despite the fact that it is an infinite sequence, the search time in practice is comparable to search with competitive finite sequences. Another important observation is that the optimal multi-agent strategy can result in significant cost reductions from a strategy that only attempts to find one agent at a time.

Finally, it is shown how increasing threshold search is applicable to *economic search* [40, 43], in which the searcher attempts to optimize a function that integrates both search costs and the value of the agent ultimately found. Search theory is an important research domain, flourishing in many disciplines, and best known perhaps for its applications to labor markets, marriage markets, monetary economics, and information theory [37, 46, 64]. By finding the lowest valued agent with a minimal search cost, increasing threshold search potentially achieves this goal, and in many settings can lead to a better overall performance from the economic search point of view. It is also shown how economic search strategies can be combined with threshold-based searches to further reduce overall costs. These results are applicable to sensor networks as well. For example, if all sensors contain data that is of equal value to the application, then the application should obtain the data with the lowest transmission cost. Since searching for this data incurs a transmission cost itself, the optimal strategy is actually a compromise between the transmission cost of the data finally obtained and the cost of finding that data.

Chapter 5 discusses the solutions presented, their similarities and differences, and further issues for consideration. The appendices present proofs and derivations of solutions from Chapter 2 and a table of notations used in Chapter 4.

Chapter 2

Zone Allocation

The chapter begins in Section 2.1 by defining the assumptions, notation, and formulas used in the analysis. Section 2.2 analyzes the problem for a field with two zones, presenting formulas for the optimal allocation, the minimum sensor count required for any expected level of coverage, and an upper bound on the maximum improvement in coverage over a baseline strategy that is oblivious to the difference in coverage between zones. Section 2.3 analyzes the problem for a field with multiple zones. Both Sections 2.2 and 2.3 present several search strategies for the optimal allocation, which can be used for more general deployment and coverage models and are used to find the optimal allocation experimentally in Section 2.5. Section 2.4 describes how to extend these solutions some other deployment and coverage models. Section 2.5 presents the results of simulations designed to test the assumptions upon which the analysis is constructed. Appendix A contains a proof of Proposition 2.8 in Section 2.3. Derivations of select solutions can be

found in Appendix B.

2.1 Preliminaries

The analysis here is for area coverage by the random uniform distribution of sensors in each zone. Sensing coverage is evaluated according to the binary sensing model, in which a sensor detects all events in its sensing area. The only assumption about sensing areas are that they have the same area within each zone; however, the shapes of sensing areas may vary between and even within zones. Since the actual assignment of sensor locations is random, only *expected coverage* can be measured. Expected coverage is simply referred to as *coverage*, and the complement of coverage is referred to as *exposure*. Coverage and exposure are defined as the *fraction* of the area covered and not covered, respectively, by a sensor network.

An *allocation* is the partition of sensors to be distributed to the different zones. In the *oblivious allocation*, the fraction of sensors allocated to each zone equals the fraction of the field the zone covers. Since this is the most reasonable strategy if no information about sensing areas is available, it is used as a base for comparing other strategies. *Absolute improvement* is the difference in coverage between two allocations. *Relative improvement* is the ratio of coverage by two different allocations.

All measurements are made in arbitrary units. The total area of the sensor field is A, the maximum sensing area of any sensor is S, and the total number of sensors

Table 2.1: Parameters for m zones

Zone	Zone area	Sensing area	Sensor count
Z_1	$\gamma_1 A$	$\alpha_1 S$	$n_1 = \beta_1 n$
		...	
Z_m	$\gamma_m A$	$\alpha_m S$	$n_m = \beta_m n$

deployed is n. C refers to the fractional area coverage. Without loss of generality, the field is partitioned into m zones $(Z_1, Z_2, \ldots, Z_m)$ such that, for each zone Z_i, the area is $\gamma_i A$, $\sum_{i=1}^{m} \gamma_i = 1$; the sensing area in it is $\alpha_i S$, $1 = \alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_m > 0$; and the number of sensors allocated to it is $n_i = \beta_i n$, $\sum_{i=1}^{m} \beta_i = 1$. See Table 2.1 for a summary. According to this formulation, an allocation in which $\beta_i = \gamma_i$ is the oblivious allocation. The order $1 = \alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_m > 0$ is imposed for the sake of solutions that required an ordering of the zones by sensing area.

In general, the allocation problem can be defined as follows: Given m functions $f_i(n_i)$ that characterize the quality of coverage of Z_i by n_i sensors, for all $1 \leq i \leq m$, and $g(f_1(n_1), \ldots, f_m(n_m))$ that characterizes the coverage of an m-zone field by an allocation $(n_1, n_2, \ldots, n_m)$ of n sensors, find an allocation that maximizes g. This paper specifically analyzes the case when $f_i(x)$ characterizes expected coverage by a random distribution of sensors in Z_i. The exact formulation of $f_i(x)$ is the subject of the following discussion.

Liu and Towsley [41] cite a result in stochastic geometry [21] that the expected coverage of the infinite plane is $1 - e^{-\lambda S}$, where S is the expected sensing area and λ is the expected number of sensors per unit area. This is for the case when sensor positions are modeled by a stationary two-dimensional Poisson point process with density λ. This formula can be modified to calculate the coverage of a bounded

region with area A by n uniformly distributed sensors *located within the field* [36]:

$$1 - e^{-\frac{Sn}{A}} \tag{2.1}$$

Lazos and Poovendran [36] derive a more accurate formula for coverage by n uniformly distributed *heterogeneous* sensors that *intersect a field*, which accounts for border effects and the shapes of individual sensing areas:

$$1 - \prod_{i=1}^{n} \left(\frac{2\pi F_0 + L_0 L_i}{2\pi(F_0 + F_i) + L_0 L_i} \right) \tag{2.2}$$

Here, F_0 is the field area, L_0 is the field perimeter, and F_i and L_i are the area and perimeter, respectively, of each sensor i. Koskinen [30] provides a less precise formula:

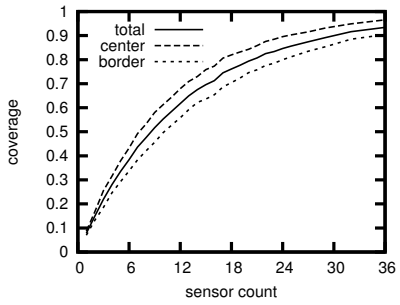
$$1 - (1 - S/A')^n \tag{2.3}$$

Here, A' is the area of the entire region that a sensor placed within it will intersect the field. Both Eq. (2.2) and Eq. (2.3) equal Eq. (2.1) in the limit as the area of the field approaches infinity.

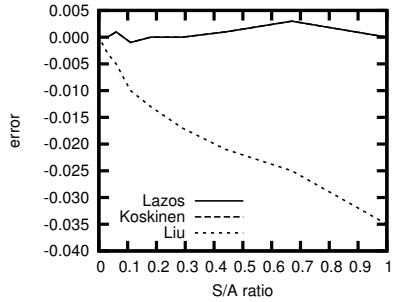
Although Eq. (2.2) accounts for differing coverage along the border of the field, it is not a measure of coverage by sensors located strictly within the field, and the assumption here is that sensors only affect coverage of the field in which they are located. Koskinen [30] provides an exact formula for the coverage of a disk-shaped field by disk-shaped sensors placed within the field, but this includes an unsolved integral and is only applicable to that scenario. As demonstrated in

the following sections, Eq. (2.1) leads to a simple analysis that yields fairly good results in practice (equations Eq. (2.2) and Eq. (2.3) can easily be substituted for Eq. (2.1) by using a different value for S/A in $e^{-Sn/A}$).

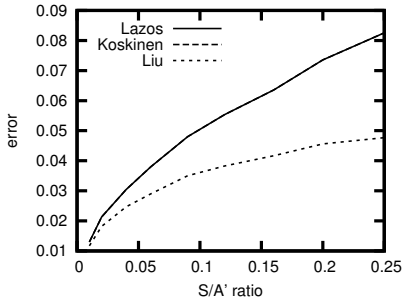
It remains to be shown how effective these formulas are as estimates of expected coverage by sensors placed *within* a bounded region. Figure 2.1 shows the results of simulations using the synthetic environment described in Section 2.5. In these simulations, disk-shaped sensors with radii ranging from 10 to 50 units were placed in a disk-shaped field with a 100 unit radius. The coverage of the entire field (whose area is denoted A'); the center (whose area is denoted A), defined as the disk centered at the field center with radius 100 minus the sensing radius; and the border, which is the remaining portion of the field, were measured for sensor counts ranging from one to the number needed for an expected coverage of 0.98 according to Eq. (2.1). Figure 2.1(a) shows the results for sensing range 30, for which the ratio S/A' equals 0.09 and the required number of sensors for an expected coverage of 0.98 is 36. Note the disparity between coverage of the center and the border, which is as high as 0.12. Figure 2.1(b) shows the difference between the simulated coverage of the center and the expected coverage according to Eq. (2.1), Eq. (2.2), and Eq. (2.3), over all sensor counts. Note that expected coverage according to Eq. (2.2) and Eq. (2.3) are equal in this case, so only the plot for the difference between simulated coverage and expected coverage according to Eq. (2.2) is visible. Here it is clear that Eq. (2.2) and Eq. (2.3) are indeed very good estimates of coverage, while Eq. (2.1) is a fair estimate, with error exceeding 0.02 only when $S/A > 0.4$. The results are not as good when border



(a) coverage by sensor size $\pi 30^2$



(b) center coverage



(c) field coverage

Figure 2.1: Coverage of a disk-shaped field

effects are considered, as shown in Figure 2.1(c). Already when $S/A' = 0.09$, the errors of Eq. (2.2) and Eq. (2.3) are close to 0.05, while the error of Eq. (2.1) is 0.035. It is clear that Eq. (2.1) is a better estimate of coverage in this case, with an acceptable error of 0.025 when $S/A' \leq 0.05$.

Based on the observations above, Eq. (2.1) is the foundation for all solutions in this paper. Consequently, Eq. (2.4) can be used to determine the number of sensors required for expected coverage C.

$$n = -\ln(1-C)\frac{A}{S} \tag{2.4}$$

2.2 Two Zones

This section addresses the specific case of allocating sensors in a field with only two zones, Z_1 and Z_2. To simplify the derivations of the solutions, a single value β is used to define the sensor allocation, such that $\beta_2 = \beta$ and $\beta_1 = (1-\beta)$. Additionally, α_1 is included in most of the expressions below, even though it can be excluded, to correlate the results here with those in Section 2.3. Based on Eq. (2.1), coverage can be calculated in $O(1)$ operations with the following formula (it is assumed that each arithmetic operation can be performed in constant time):

$$\begin{aligned} &\gamma_1\left(1-e^{-\frac{\alpha_1 S}{\gamma_1 A}(1-\beta)n}\right)+\gamma_2\left(1-e^{-\frac{\alpha_2 S}{\gamma_2 A}\beta n}\right) \\ &=1-\gamma_1 e^{-\frac{\alpha_1 S}{\gamma_1 A}(1-\beta)n}-\gamma_2 e^{-\frac{\alpha_2 S}{\gamma_2 A}\beta n} \end{aligned} \tag{2.5}$$

This reduces to Eq. (2.1) when $\beta = \gamma_2$ and $\alpha_2 = 1$. The optimal allocation can be derived using exhaustive search in $O(n)$ time or binary search (see Algorithm 1 for one implementation) in $O(\log n)$ time. Binary search is valid because Eq. (2.5), as a function of β, is concave in the interval $[0,1]$; consequently, there is only one maximum in coverage. This can be shown by taking the second derivative of Eq. (2.1) with respect to β.

These search strategies are applicable even when coverage is not characterized by Eq. (2.5), although binary search is limited to cases in which there is a single maximum in coverage. They will be useful for experimentally finding the optimal allocation in Section 2.5. When coverage is characterized by Eq. (2.5), however,

Algorithm 1 *BinarySearch*(n) returns $(n', n-n')$

```
l ← 0; r ← n
while l < r do
  m1 ← l + 1/4(r − l); m2 ← l + 1/2(r − l); m3 ← l + 3/4(r − l)
  a1 ← (m1, n − m1)
  a2 ← (m2, n − m2)
  a3 ← (m3, n − m3)
  if coverage by a1 > coverage by a2 then
    r ← m2 − 1
  else if coverage by a2 > coverage by a3 then
    l ← m1 + 1; r ← m3 − 1
  else
    l ← m2 + 1
  end if
end while
return (l, n − l)
```

the optimal allocation can be derived in constant ($O(1)$) time with Solution 2.1.

Solution 2.1 (Direct calculation). Calculate n_2 by first calculating β_{opt} with Eq. (2.6) and then rounding $\beta_{opt} n$ to the integer for which coverage is maximum.

$$\beta_{opt} = \begin{cases} 0 & \text{if } n \leq -\frac{A}{S}\frac{\gamma_1}{\alpha_1}\ln\alpha_2 \\ \frac{1}{\sum_{i=1}^{2}\frac{\gamma_i}{\alpha_i}}\frac{\gamma_2}{\alpha_2}\left(\frac{A}{Sn}\frac{\gamma_1}{\alpha_1}\ln\alpha_2 + 1\right) & \text{otherwise} \end{cases} \tag{2.6}$$

Proof. Since there is only one maximum in coverage, it is sufficient set the derivative of Eq. (2.5) with respect to β equal to 0 and solve for β. See Appendix B.1 for the complete derivation. For values of n for which the resulting formula is less than 0, set β equal to 0. □

Based on Solution 2.1, the number of sensors required to guarantee a certain

level of coverage can be calculated directly using the next solution.

Solution 2.2. The number of sensors n required to achieve coverage C under the optimal allocation can be calculated as follows:

$$n = \begin{cases} -\frac{\gamma_1 A}{\alpha_1 S} \ln \frac{\gamma_1 - C}{\gamma_1} & \text{if } C < \gamma_1(1-\alpha_2) \\ \sum_{i=1}^{2} \frac{\gamma_i}{\alpha_i} \frac{A}{S} \left(\ln(1-C) - \ln\left(\sum_{i=1}^{2} \frac{\gamma_i}{\alpha_i} \right) - \frac{1}{\sum_{i=1}^{2} \frac{\gamma_i}{\alpha_i}} \frac{\gamma_2}{\alpha_2} \ln \alpha_2 \right) & \text{otherwise} \end{cases} \tag{2.7}$$

Proof. First derive Eq. (2.8), the formula for coverage C by the optimal allocation of n sensors, by substituting Eq. (2.6) for β in Eq. (2.5), and then solve for n. See Appendix B.2 and B.3 for the complete derivations.

$$C = \begin{cases} \gamma_1 \left(1 - e^{-\frac{\alpha_1 S}{\gamma_1 A} n} \right) & \text{if } n \leq -\frac{A}{S} \frac{\gamma_1}{\alpha_1} \ln \alpha_2 \\ 1 - \sum_{i=1}^{2} \frac{\gamma_i}{\alpha_i} \left(\alpha_2^{\frac{\gamma_2}{\alpha_2}} \right)^{\frac{1}{\sum_{i=1}^{2} \frac{\gamma_i}{\alpha_i}}} e^{\frac{1}{\sum_{i=1}^{2} \frac{\gamma_i}{\alpha_i}} \frac{Sn}{A}} & \text{otherwise} \end{cases} \tag{2.8}$$

□

Finally, Solution 2.3 upper bounds the improvement of Solution 2.1 over the oblivious allocation, which is to set $\beta = \gamma_2$. The upper bound depends on the value of β_{opt}. When $\beta_{opt} < \gamma_2$, the maximum absolute improvement is trivially 0.5 when $S \to A$, $\gamma_2 \to 0$, $\alpha_2 \to 0$, and only one sensor is deployed. In less extreme cases, significant coverage is only achieved when the sensor count is large enough such that $\beta_{opt} > \gamma_2$. Therefore, the focus here is on the case when $\beta_{opt} > \gamma_2$.

Solution 2.3. An upper bound in absolute improvement when $\beta_{opt} > \gamma_2$, for any set of values of α_2, γ_2, n, A, and S, can be calculated with the following formula.

$$\gamma_1\gamma_2(1-\alpha_2)(\gamma_1\alpha_2+\gamma_2)^{\frac{\gamma_1\alpha_2+\gamma_2}{\gamma_1(1-\alpha_2)}}\alpha_2^{\frac{\alpha_2}{1-\alpha_2}} \tag{2.9}$$

Proof. First, observe that β increases with n, since $\ln\alpha_2$ is negative for $\alpha_2 < 1$ and $\lim_{n\to\infty}\frac{A\gamma_2\gamma_1\ln\alpha_2}{Sn} = 0$. Therefore, the improvement in Z_2 is an upper bound on the total improvement. Begin with the difference in coverage of Z_2 between using $\beta = \beta_{opt}$ and $\beta = \gamma_2$.

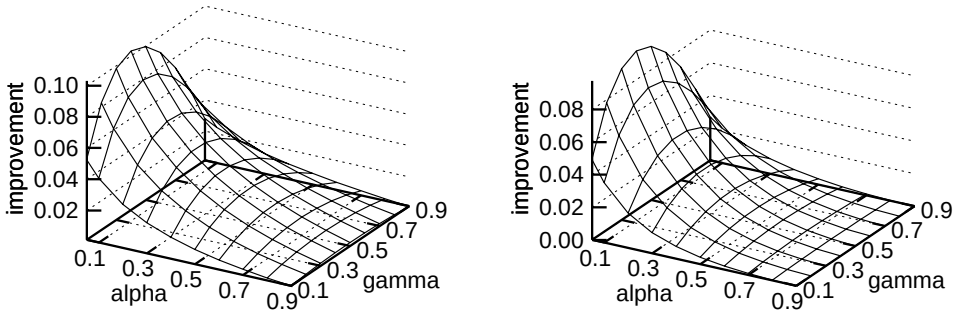
$$\gamma(e^{-\frac{\alpha_2 S}{\gamma_2 A}\gamma_2 n} - e^{-\frac{\alpha_2 S}{\gamma_2 A}\frac{1}{\gamma_1\alpha_2+\gamma_2}(\frac{A\gamma_2\gamma_1\ln\alpha_2}{Sn}+\gamma_2)n}) \tag{2.10}$$

Find the value of n at which this is maximum by setting the derivative with respect to n equal to 0 and solving for n.

$$n = -\frac{A}{\alpha_2 S}\left(\frac{\gamma_1\alpha_2+\gamma_2}{\gamma_1(1-\alpha_2)}\ln(\gamma_1\alpha_2+\gamma_2) + \frac{\alpha_2}{1-\alpha_2}\ln\alpha_2\right) \tag{2.11}$$

Substitute this value of n back into Eq. (2.10) and reduce to find the upper bound. □

Figure 2.2(a) illustrates the range of values of Eq. (2.9). The maximum improvement when accounting for coverage in Zone 1 as well, as found by numerical approximation, is slightly less, as illustrated in Figure 2.2(b). In both cases, the maximum value is 0.13, when $\alpha_2 \to 0$ and $\gamma_2 \approx 0.39$. An upper bound on relative improvement can also be found by numerical approximation. As a result, the



(a) Upper bounds in improvement (b) Expected maximum improvement

Figure 2.2: Improvement of optimal allocation over oblivious allocation in a two-zone field

following conclusion can be made:

Remark 2.4. When $\beta_{opt} > \gamma_2$, absolute improvement never exceeds 0.13 and relative improvement never exceeds 1.17.

2.3 Multiple Zones

This section addresses the general case of a field with multiple (m) zones. The area of each zone Z_i, the sensing area within, and the number of sensors allocated to it are defined in Table 2.1. Accordingly, the expected coverage of the entire field by an allocation $(\beta_1 n, \ldots, \beta_m n)$ is

$$\sum_{i=1}^{m} \gamma_i \left(1 - e^{-\frac{\alpha_i S}{\gamma_i A}\beta_i n}\right) = 1 - \sum_{i=1}^{m} \gamma_i e^{-\frac{\alpha_i S}{\gamma_i A}\beta_i n} \tag{2.12}$$

This is a generalization of Eq. (2.5). The coverage of each zone can be calculated in $O(1)$ time and of the whole field in $O(m)$ time. As in the two-zone case, the optimal allocation can be found using exhaustive search of all $\binom{n+m-1}{m-1}$ possible allocations. Exhaustive search is applicable to any function of coverage by a sensor allocation. However, this is not considered polynomial in n, since, for $n = m$, $\binom{2n-1}{n-1} \geq \frac{2^{2n-1}}{\sqrt{n}}$. Even for small m, this is intractable when n is on the order of 10^3 or more, as is the case in Section 2.5. The next three solutions can find the optimal allocation in polynomial time.

Solution 2.5 (Dynamic programming, $O(n^2m)$). Derive the optimal allocation with the following dynamic programming formulation, in which $H(i,j)$ is the maximum achievable coverage by the allocation of j sensors to zones Z_1 to Z_i, and $f_i(j)$ is the coverage of Z_i by j sensors.

$$H(i,j) = \max_{0 \leq k \leq j} \{\gamma_i f_i(k) + H(i-1, j-k)\}$$
$$H(1,j) = \gamma_1 f_i(j) \tag{2.13}$$

Solution 2.6 (Greedy allocation, $O(n \log m)$).

1. Store the zones in a heap, giving priority to zones with the maximum increase in coverage when adding a sensor.
2. While there are unallocated sensors:
   (a) Remove the top zone Z from the heap
   (b) Allocate a sensor to Z
   (c) Calculate the coverage increase by adding another sensor to Z

(d) Insert Z in the heap

Proof. A proof by contradiction shows that the greedy algorithm obtains the optimal allocation. If the final allocation is not optimal, then coverage can be improved by moving at least one sensor from Z_i to Z_j, for some i and j. Let this be the last sensor allocated to Z_i in the one-by-one allocation. The increase in coverage was greater by assigning it to Z_i rather than Z_j in the round that it was allocated to Z_i. The increase in coverage by any sensor added to Z_j afterwards must have been less than the increase by adding this sensor to Z_i. Otherwise, it would have been assigned to Z_j before them. Therefore, coverage cannot be improved by moving a sensor between zones. This is even true for moving several sensors, since the coverage increase declines with each sensor added to Z_j, while the coverage decrease grows as those same sensors are removed from Z_i.

The complexity is derived as follows. There are $O(n)$ rounds, one for each sensor. Each round requires $O(\log m)$ steps to reorder the heap. The coverage increase can be calculated in $O(1)$ time by applying Eq. (2.1). □

Solution 2.7 (Binary search, $O((logn)^{m-1})$). The idea behind this algorithm is to perform a binary search of the optimal partition of n sensors between zone Z_1 and zones $Z_2,\ldots,Z_m$, with the optimal partition of the sensors allocated to $Z_2,\ldots,Z_m$ derived recursively by the same method. Algorithm 2 is one way to implement this algorithm.

These solutions are applicable even when coverage is not characterized by Eq. (2.12). The greedy algorithm applies when each f_i is strictly increasing, while dy-

Algorithm 2 *BinarySearch*$(Z_i, n', (n_1, \ldots, n_{i-1}))$ returns $(n_1, \ldots, n_m)$

```
if i = m then
  return (n_1, ..., n_{m-1}, n')
else
  l ← 0; r ← n'
  while l < r do
    m_1 ← l + (1/4)(r − l); m_2 ← l + (1/2)(r − l); m_3 ← l + (3/4)(r − l)
    a_1 ← BinarySearch(Z_{i+1}, n' − m_1, (n_1, ..., n_{i−1}, m_1)
    a_2 ← BinarySearch(Z_{i+1}, n' − m_2, (n_1, ..., n_{i−1}, m_2)
    a_3 ← BinarySearch(Z_{i+1}, n' − m_3, (n_1, ..., n_{i−1}, m_3)
    if coverage by a_1 > coverage by a_2 then
      r ← m_2 − 1
    else if coverage by a_2 > coverage by a_3 then
      l ← m_1 + 1; r ← m_3 − 1
    else
      l ← m_2 + 1
    end if
  end while
  return Best allocation found in binary search
end if
```

namic programming is suitable for all types of functions f. Binary search is only applicable when there is a single maximum coverage of zones $Z_1, \ldots, Z-i$, for all $i \leq m$, by n sensors. These solutions will be useful in Section 2.5. However, when coverage is characterized by Eq. (2.12), the optimal allocation can be derived in $O(m)$ time with Solutions 2.10 and 2.13.

Solution 2.10 uses an approach similar to Solution 2.5: Given the optimal coverage by the allocation of any number of sensors $n' \leq n$ to zones $Z_1 \ldots Z_{i-1}$, determine the optimal partition of n sensors between Z_i and $Z_1 \ldots Z_{i-1}$. However, instead of using dynamic programming to calculate the optimal coverage and partition, these values are calculated directly in $O(1)$ time, resulting in an

$O(m)$ solution. For this solution, let Z'_i be the combination of zones $Z_1, \ldots, Z_i$ and β'_i be a partition of sensors between Z_i and Z'_{i-1}. The expected coverage is

$$1 - \sum_{i=1}^{m} \gamma_i e^{-\frac{\alpha_i S}{\gamma_i A} \beta'_i \prod_{j=i+1}^{m} (1-\beta'_j) n} \tag{2.14}$$

The following proposition establishes that the coverage of zone Z'_i by n sensors can be calculated with a single expression in $O(1)$ time. Solutions 2.10 and 2.14 follow as a direct result.

Proposition 2.8. Coverage by an optimal allocation of n sensors to zones $Z_1 \ldots Z_i$ that results in the allocation of at least one sensor to each zone is equal to the expression

$$\sum_{j=1}^{i} \gamma_j \left(1 - C_i e^{-\frac{\alpha'_i S}{\sum_{j=1}^{i} \gamma_j A} n} \right) \tag{2.15}$$

for some values of C_i and α'_i.

The proof of Proposition 2.8 can be found in Appendix A.

Note 2.9. Proofs by induction establish that Eq. (2.16) and Eq. (2.17) below are closed form expressions for α_i and C_i, respectively. These expressions, along with the equality $\sum_{j=1}^{i-1} \frac{\gamma_j}{\alpha_j} \ln \alpha_i - \sum_{j=1}^{i-1} \frac{\gamma_j}{\alpha_j} \ln \alpha_j = \sum_{j=1}^{i} \frac{\gamma_j}{\alpha_j} \ln \alpha_i - \sum_{j=1}^{i} \frac{\gamma_j}{\alpha_j} \ln \alpha_j$, are used in simplifying Eq. (2.15) and Eq. (A.2), as found in Solutions 2.14 and 2.10,

respectively.

$$\alpha'_i = \frac{\sum_{j=1}^{i} \gamma_j}{\sum_{j=1}^{i} \frac{\gamma_j}{\alpha_j}} \tag{2.16}$$

$$C_i = \frac{\sum_{j=1}^{i} \frac{\gamma_j}{\alpha_j}}{\sum_{j=1}^{i} \gamma_j} \left(\prod_{j=1}^{i} \alpha_j^{\frac{\gamma_j}{\alpha_j}} \right)^{\frac{1}{\sum_{j=1}^{i} \frac{\gamma_j}{\alpha_j}}} \tag{2.17}$$

Solution 2.10 (Recursive allocation, $O(m)$). Begin by calculating β'_{opt_m} with Eq. (2.18) below, rounding $\beta'_{opt_m} n$ to the integer for which coverage is largest. Allocate this number of sensors to Z_m, and then recursively partition the remaining sensors in Z'_{m-1} using the same method.

$$\beta'_{opt_i} = \begin{cases} 0 & \text{if } n < -\frac{A}{S} \left(\sum_{j=1}^{i-1} \frac{\gamma_j}{\alpha_j} \ln \alpha_i - \sum_{j=1}^{i-1} \frac{\gamma_j}{\alpha_j} \ln \alpha_j \right) \\ \frac{\gamma_i}{\alpha_i} \frac{1}{\sum_{j=1}^{i} \frac{\gamma_j}{\alpha_j}} \left(\frac{A}{Sn} \left(\sum_{j=1}^{i-1} \frac{\gamma_j}{\alpha_j} \ln \alpha_i - \sum_{j=1}^{i-1} \frac{\gamma_j}{\alpha_j} \ln \alpha_j \right) + 1 \right) & \text{otherwise} \end{cases} \tag{2.18}$$

The complete details of the derivation Eq. (2.18) can be found in Appendices A and B.4.

Note 2.11. When $m = 2$, this equation equals Eq. (2.6).

Note 2.12. The oblivious allocation is to set $\beta'_i = \gamma_i / \sum_{j=1}^{i} \gamma_j$.

Alternatively, the number of sensors required to guarantee a certain level of coverage can be calculated directly using the next solution.

Solution 2.13 (Direct calculation, $O(m)$). First, determine the largest k for which

$$n > -\frac{A}{S}\left(\sum_{j=1}^{k-1}\frac{\gamma_j}{\alpha_j}\ln\alpha_k - \sum_{j=1}^{k-1}\frac{\gamma_j}{\alpha_j}\ln\alpha_j\right)$$

Then allocate sensors according to the following partition:

$$\beta_{opt_i} = \begin{cases} 0 & \text{if } i > k \\ \frac{\gamma_i}{\alpha_i}\frac{1}{\sum_{j=1}^{k}\frac{\gamma_j}{\alpha_j}}\left(\frac{A}{Sn}\left(\sum_{j=1}^{k-1}\frac{\gamma_j}{\alpha_j}\ln\alpha_i - \sum_{j=1}^{k-1}\frac{\gamma_j}{\alpha_j}\ln\alpha_j\right)+1\right) & \text{otherwise} \end{cases} \tag{2.19}$$

Proof. This is derived using the method of Lagrange multipliers. A sketch of the derivation is as follows. Set the Lagrange function to

$$1 - \sum_{i=1}^{k}\gamma_i e^{-\frac{\alpha_i S}{\gamma_i A}\beta_i n} - \lambda\left(\sum_{i=1}^{k}\beta_i - 1\right) \tag{2.20}$$

Set the partial derivative with respect to each β_i equal to 0 and solve for each β_i, such that, for all $1 \le i \le k$,

$$\beta_i = -\frac{\gamma_i A}{\alpha_i S n}\left(\ln\frac{\lambda A}{Sn} - \ln\alpha_i\right) \tag{2.21}$$

Take the partial derivative of the Lagrangian with respect to λ and set it equal to 0, substitute Eq. (2.21) for each β_i, and solve for $\ln\frac{\lambda A}{Sn}$.

$$\ln\frac{\lambda A}{Sn} = -\frac{1}{\sum_{j=1}^{k}\frac{\gamma_j}{\alpha_j}}\left(\frac{Sn}{A} - \sum_{j=1}^{k}\frac{\gamma_j}{\alpha_j}\ln\alpha_j\right) \tag{2.22}$$

Substitute this back into Eq. (2.21) and simplify to get Eq. (2.19). Note that k should be set as large as possible such that $\beta_i > 0$, for all $1 \leq i \leq k$.

□

Similar to Solution 2.2, the number of sensors required to guarantee a certain level of coverage can be calculated directly using the next solution.

Solution 2.14. The number of sensors n required to achieve coverage C under the optimal allocation can be calculated as follows:

$$n = \frac{A}{S}\sum_{j=1}^{i}\frac{\gamma_j}{\alpha_j}\left(\ln\left(\sum_{j=1}^{i}\gamma_j - C\right) - \ln\sum_{j=1}^{i}\frac{\gamma_j}{\alpha_j} - \frac{1}{\sum_{j=1}^{i}\frac{\gamma_j}{\alpha_j}}\sum_{j=1}^{i}\frac{\gamma_j}{\alpha_j}\ln\alpha_j\right) \quad (2.23)$$

where $i = \text{argmax}_{i\geq 1}\left(C \geq \sum_{j=1}^{i}\gamma_j - \sum_{j=1}^{i}\frac{\gamma_j}{\alpha_j}\alpha_i\right)$

Proof. According to Proposition 2.8, optimal coverage of the entire field is

$$C = \sum_{j=1}^{i}\gamma_j\left(1 - C_i e^{-\frac{\alpha_i' S}{\sum_{j=1}^{i}\gamma_j A}n}\right) \quad (2.24)$$

where $i = \underset{i\geq 1}{\text{argmax}}\left(n \geq -\frac{A}{S}\left(\sum_{j=1}^{i-1}\frac{\gamma_j}{\alpha_j}\ln\alpha_i - \sum_{j=1}^{i-1}\frac{\gamma_j}{\alpha_j}\ln\alpha_j\right)\right)$. Optimal coverage of the entire field is therefore

$$\sum_{j=1}^{i}\gamma_j\left(1 - \frac{\sum_{j=1}^{i-1}\frac{\gamma_j}{\alpha_j}}{\sum_{j=1}^{i-1}\gamma_j}\alpha_i\right) \quad (2.25)$$

when $n = -\frac{A}{S}\left(\sum_{j=1}^{i-1}\frac{\gamma_j}{\alpha_j}\ln\alpha_i - \sum_{j=1}^{i-1}\frac{\gamma_j}{\alpha_j}\ln\alpha_j\right)$ □

2.4 Extensions

This section shows how the generic solutions above can be used for deterministic deployment. It also shows how the $O(m)$ solutions can be applied to two general models of coverage.

2.4.1 Deterministic deployment

The optimal arrangement of sensors in a deterministic deployment varies with the number of sensors, the size and shape of their sensing areas, and the size and shape of the field of coverage. Unfortunately, there are no general results for these types of problems [6]. However, it is possible to make the following assumption to guide the allocation process, at least in the case of disk-shaped sensing areas: Assume that the optimal arrangement in a bounded region is the same as that for an unbounded region, which is along the grid points of a triangular lattice, such that each sensor occupies a hexagonal area. The coverage of a region with area A by n sensors is therefore equal to the coverage of any hexagon a sensor occupies. The size of such a hexagon is A/n, such that the distance d between sensors is $\sqrt{\frac{2A}{\sqrt{3}n}}$. The radius r of a sensing area with size S is $\sqrt{S/\pi}$. The optimal allocation can be derived with the greedy algorithm, using the following formula for coverage [48]:

$$C = \begin{cases} 1 & \frac{d}{r} \leq \sqrt{3} \\ 1 - \frac{A}{\frac{\sqrt{3}}{4}d^2} & \sqrt{3} < \frac{d}{r} \leq 2 \\ \frac{2\pi}{\sqrt{3}}\left(\frac{r}{d}\right)^2 & \frac{d}{r} > 2 \end{cases} \tag{2.26}$$

where

$$A = \frac{\sqrt{3}}{4}B^2 - 3r^2 \arcsin\frac{B}{2r} + \frac{3}{4}B\sqrt{4r^2 - B^2}$$
$$B = \frac{d}{2} - \sqrt{3r^2 - \frac{3}{4}d^2}$$

2.4.2 Fault tolerance

Fault tolerance is an important issue in sensor network deployment [52], as power supplies are limited and sensors are prone to failure even upon deployment. In this case, if all sensors in zone Z_i are associated with a probability of failure p_i, then Solutions 2.10 and 2.13 can still be applied by replacing α_i with $p_i\alpha_i$ and reordering the zones according to the values of $p_i\alpha_i$.

2.4.3 Differentiated detection

Many works consider regions of preferential coverage within the deployment field [15, 68], such as densely populated areas [66] and amphibian hot-spots [53]. One way to address this issue is by weighting the coverage of each zone according to its priority, with the objective of maximizing weighted coverage [66]. In this case, Eq. (2.12) can be modified by multiplying the coverage of each zone by weight w_i rather than γ_i. The weights can be normalized such that they sum to one and maximum coverage is upper bounded by one. Coverage is calculated with the

following modification of Eq. (2.12):

$$\sum_{i=1}^{m} w_i \left(1 - e^{-\frac{\alpha_i S}{\gamma_i A}\beta_i n}\right) \tag{2.27}$$

The optimal allocation can be derived with Solution 2.13, except that $\ln \alpha_i$ is replaced with $\ln \frac{w_i \alpha_i}{\gamma_i}$ and the zones are ordered according to the value of $\frac{w_i \alpha_i}{\gamma_i}$.

2.5 Simulation Results

This section evaluates the accuracy of the above solutions through simulations. Specifically, optimal coverage for various scenarios is found experimentally–using simulated coverage in a binary search–and compared to coverage by the analytically derived optimal allocation. The results are very encouraging: Even though expected coverage differs from simulated coverage, the analytic optimal allocation is close to the experimentally optimal allocation. This means that the formulas presented in Section 2.1 can be used effectively to derive the optimal allocation, despite their imprecision. The improvement in coverage by the optimal allocation over the oblivious allocation and other base strategies is reported as well. Finally, zone allocation for deterministic deployment is compared to other strategies.

In the simulations, sensing ranges are calculated using a weighted distance function [14] to model sensing areas more realistically than the standard disk sensing model. According to this model, a line between any two points in the field is partitioned into line segments by the zones of the field. The weighted distance

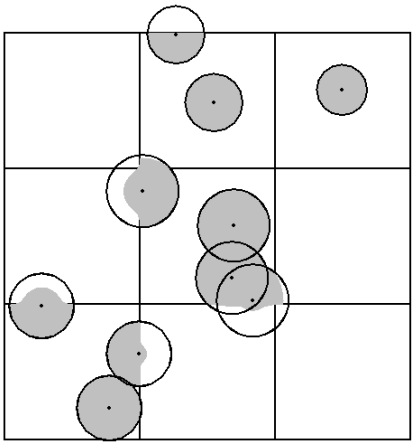


Figure 2.3: Coverage under the weighted distance model visualized

between the points is the sum of the segment lengths normalized by the sensing range in the zones to which they belong. For example, if the line from A to point B covers 10 units, 15 units, and 5 units in zones Z_1, Z_2, and Z_3, respectively, and the sensing range in Z_1, Z_2, and Z_3 are respectively 40, 30, and 20 units, then the weighted distance between A and B is $10/40 + 15/30 + 5/20 = 1$. A sensor covers all points whose weighted distance from the sensor is less than one. The sensing area of a sensor is disk-shaped when all of the points it covers are in the same zone in which the sensor is located. The values of S and α_i are set according the areas of these disk shapes. In the example provided, $S = \pi 40^2$, $\alpha_2 = \left(\frac{3}{4}\right)^2$, and $\alpha_3 = \left(\frac{1}{2}\right)^2$. Figure 2.3 shows how the sensing areas are affected by this model in a field with nine zones arranged in a 3x3 grid. The circles indicate what the sensing areas would be without the weighted distance model.

In order to simplify the problem of calculating the area covered, coverage is measured by superimposing a square lattice on the field of deployment and

counting the number of points that are covered by at least one sensor. For example, coverage of a 400 unit by 400 unit field is measured by superimposing a $400x400$ grid on the field, with a distance of one unit between neighboring points. In order to further simplify calculations, sensor locations are restricted to lattice points. Simulations in select scenarios showed that the standard deviation *of the mean* of fifty repetitions is 0.003, implying that the probability that the average of fifty simulations differs from the actual mean by less than 0.005 is 95%. While this is satisfactory to estimate coverage by an allocation, it is not satisfactory for any search method. Small errors in coverage estimates can lead to large errors in the search process; in fact, greedy allocation is ineffective because it provides very bad partitions. Binary search is better, but still results in some error. In order to reduce the search error, the average of five hundred simulations is used instead.

The first set of results are for a two zone field with area $A = 400^2$, a 20 unit sensing range in Z_1, α_2 ranging from 0.05 to 0.90, γ_2 ranging from 0.1 to 0.9, and sensor counts ranging from 1 to the number required for expected coverage of 0.98. Simulated coverage differed from expected coverage by as much 0.02 for both the oblivious and optimal allocations. Optimal coverage could not be determined with complete accuracy; for example, coverage by the analytically optimal allocation was sometimes better than the experimentally optimal allocation, even if the allocations were the same. However, coverage by both methods never differed by more than 0.003. This indicates that the analytic solution is close to optimal, even though it does not account for the border effects that were present in the simulations.

Figure 2.4 shows the maximum improvement in coverage by the optimal allocation over the oblivious allocation when $\beta_2 > 0$. It is similar to Figure 2.2(b), but the values sometimes differ by as much as 0.008. This is because coverage is not accurately represented by the formulas used to derive the solutions. For example, Figure 2.5 shows the difference in expected and simulated coverage by the oblivious and analytically optimal allocations when $\alpha = 0.2$ and $\gamma = 0.5$. The difference is greater for the optimal allocation than for the oblivious allocation. Accordingly, the expected improvement is greater than the actual improvement. Finally, the oblivious allocation was compared to a uniform distribution of sensors throughout the field, since practically it may be simpler to use a uniform distribution than some allocation scheme. The difference in coverage was at most 0.007 and 0.001 on average.

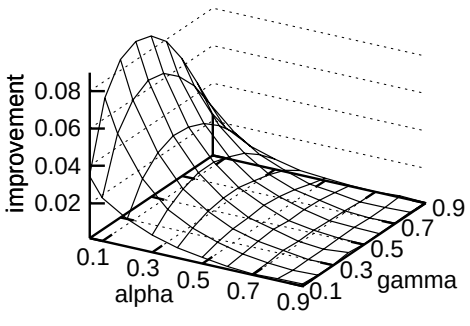


Figure 2.4: Maximum improvement in simulated coverage of a two-zone field

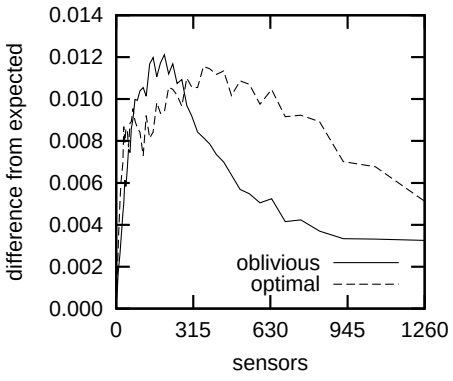


Figure 2.5: Difference between expected and simulated coverage of a two-zone field

The optimal allocation was derived experimentally for a four zone field as well. The area of the field area was $A = 400^2$, the sensing ranges were 8, 12,

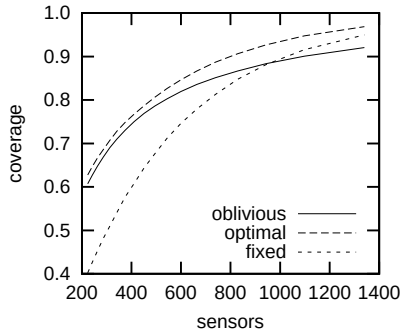


Figure 2.6: Coverage of a nine zone field

16, and 20, and each zone occupied a different quadrant of the field. Finding the optimal solution is challenging in such a scenario. On average, coverage by the experimental allocation was 0.00009 less than coverage by the analytic allocation, but still no less than 0.0035. The number of allocations evaluated during a binary search is best represented by the function $(1.02\log n)^4$. 12,248 allocations were evaluated to find the optimal allocation of 1,147 sensors–the number of sensors required for coverage of 0.96. Based on this estimate, the number of allocations that need to be evaluated becomes intractable even for a field with nine zones. For this reason, the analytic allocation is the best method for finding the optimal solution. Figure 2.6 shows coverage of a nine zone field, in which $A = 900^2$ and sensing ranges were $8, 12, \ldots, 40$, by the oblivious and optimal allocations and a new allocation called "fixed". According to this strategy, β'_{opt_i} is set to $\frac{\gamma_i}{\alpha_i}\frac{1}{\sum_{j=1}^{i}\frac{\gamma_j}{\alpha_j}}$, which is the value of β'_{opt_i} in Eq. (2.18) as $n \to \infty$. The fixed strategy is meant to be an alternative to the oblivious allocation that also uses fixed proportions. It is clear that it is a good strategy for large sensor counts only.

Finally, deterministic deployment strategies were compared for the four zone field described above. Figure 2.7 shows coverage by three strategies: uniform placement along a triangle lattice superimposed on the entire field; zone allocation and lattice arrangement in each zone, as described in Section 2.4; and greedy assignment [15, 66, 68], in which each sensor is placed where coverage increase is greatest. Additionally, an upper bound on coverage by zone allocation is shown. Greedy assignment slightly outperformed zone allocation, and both greatly outperformed uniform placement. However, greedy assignment took over 30,000 times longer to execute than zone allocation, on average. Figure 2.8 shows the arrangement of 350 sensors by both methods. Zone allocation is more organized; better coverage is possible by changing the horizontal and vertical distances between grid points.

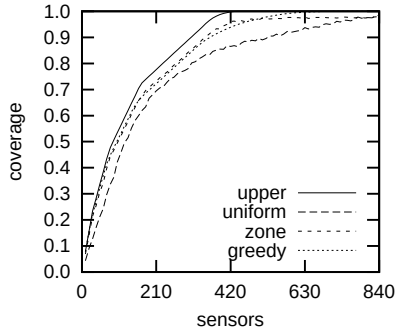
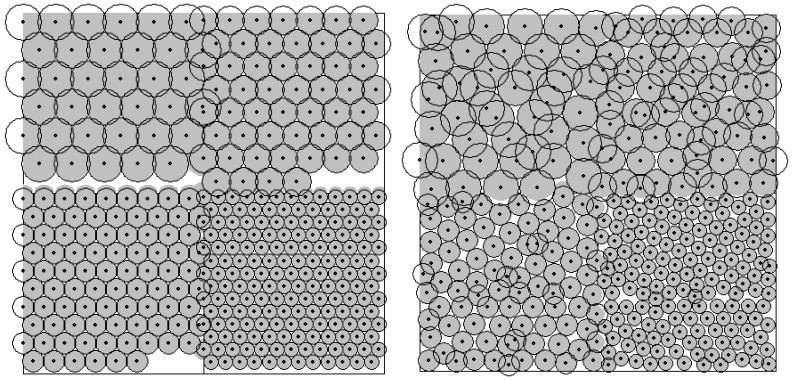


Figure 2.7: Coverage by deterministic deployment



(a) zone allocation (b) greedy assignment

Figure 2.8: Examples of Deterministic Deployment

Chapter 3

Sensor Selection

The goal of this chapter is to design a *link- and data-driven* technique to determine which sensors are most critical for overall sensing quality, especially when the real-time budget constraints are tight. In Section 3.1, we introduce the overall sensor selection problem in the context of linked information networks. In Section 3.2, we study the problem of link error modeling, which is required in order to assign values to the variables of the optimization problem. In Section 3.3, we show that the problem is NP-hard, and design effective integer programming and approximation algorithms. Section 3.4 presents the experimental results.

3.1 The Sensor Selection Problem

In this section, we introduce the sensor selection problem in the context of linked information networks. We note that the network of relationships between sen-

sors can be represented as a graph. In general, let the simple directed graph $G = (V,E)$ represent a network of n sensors and m logical links between sensors. Each vertex $v_i \in V$ represents a sensor and its associated stream, and each edge $e_{ij} = (v_i, v_j) \in E$ represents a logical link between sensors. These logical links can be designed on the basis of proximity or any other domain-specific property that affects predictions between sensors. In general, regression modeling [1] can be used to predict one sensor time series from another selected sensor. We note that it is computationally impractical to examine the behavior of all sensor pairs in order to make decisions about which sensors to select for the prediction process. For example, a network of 10^4 sensors contains 10^8 possible vertex pairs. Such a large number of possible node pairs can make the sensor selection problem computationally impractical. The logical links between sensors provide *external domain-specific information* which can *significantly reduce the computation required* in the selection process.

Each vertex v_i is associated with a cost of selection c_i, an importance u_i, and a prediction error q_i. Each edge e_{ij} is associated with a prediction error p_{ij}. We note that c_i and u_i are set in an application-specific manner. The cost c_i may reflect the consumption of power or some other resource by a sensor during the data collection process. The importance u_i reflects the importance of a sensor in providing useful information to the underlying application. The error of the regression when using v_i to predict v_j is denoted p_{ij}. The error of not predicting v_i at all is denoted q_i. The problem of selecting the optimal set of sensors in order to make predictions about all other sensors can be formulated either as a *minimization problem*

or a *maximization problem*. In the minimization formulation, we minimize the weighted error of selecting a particular set of sensors. In the maximization formulation, we maximize the weighted *reduction* in error that arises from selecting a particular set of sensors over the trivial solution of not selecting any sensors at all.

The objective in the first (minimization) formulation of the problem is to select a set of sensors $S \subseteq V$ which minimizes the total prediction error of all streams and whose total cost does not exceed some budget constraint B. In this case, we define the *weighted non-predictability error* z_i as the importance-weighted error of not being able to predict sensor v_i at all; therefore, z_i is set to $u_i \cdot q_i$. Similarly, we create a weighted version of the link prediction error, which is set according to the importance u_j of the target sensor v_j. The weighted error w_{ij} of predicting sensor v_j from sensor v_i is set to $w_{ij} = u_j \cdot p_{ij}$. The idea here is that the error of predicting sensor v_j from sensor v_i is *magnified* by the importance u_j of sensor v_j. The objective function for a set of sensors S is calculated by iterating over each node v_j and computing its contribution to the total prediction error as follows: **(1)** If v_j is in S, it contributes 0 to the error. **(2)** If v_j is not in set S, but there is an edge from at least one vertex in S to v_j, then the error it contributes is the minimum weighted prediction error $w_{ij} = u_j \cdot p_{ij}$ of sensor v_j from any linked sensor $v_i \in S$. The implicit assumption here is that a one-to-one regression model is used from sensor v_i to sensor v_j to predict the stream in sensor v_j from the stream in sensor v_i. **(3)** If v_j is not covered by any linked sensors, then the error it contributes is the weighted error $z_j = u_j \cdot q_j$ of not predicting it at all.

In the second (maximization) formulation of the problem, the objective is to select a set of sensors $S \subseteq V$ that maximizes the total *reduction* in error and whose total cost does not exceed some budget constraint B. This is the dual of the minimization problem, since maximizing the total reduction in error is equivalent to minimizing the total error. The key difference is in the definition of the edge weights. The weight of edge e_{ij} is denoted by w'_{ij} and reflects the importance-weighted error *reduction* of sensor v_j when the stream at sensor v_i is used to predict it. Specifically, w'_{ij} is set to $u_j \cdot (q_j - p_{ij})$. We note that the implicit assumption here is that $p_{ij} \leq q_j$, since q_j represents the maximum error of sensor v_j, which would arise when sensor v_j is not predicted at all. The idea is that if v_i could accurately estimate v_j in the first (minimization) formulation, then v_i eliminates all error in estimating v_j in the second (maximization) formulation, such that $w'_{ij} = z_j$. Now the objective function is computed by summing the error reductions over all vertices $v_j \in V$ as follows: **(1)** If v_j is in S, it contributes $z_j = u_j \cdot q_j$ to the error reduction. **(2)** If v_j is not in set S, but there is an edge from at least one vertex in S to v_j, then the error reduction it contributes is the maximum weighted prediction error reduction $w'_{ij} = u_j \cdot (q_j - p_{ij})$ of sensor v_j from any linked sensor $v_i \in S$. **(3)** If v_j is not covered by any linked sensors, then it does not contribute anything to error reduction. Therefore, its contribution is 0.

3.2 Link Error Modeling

In order to use the aforementioned formulations, it is necessary to model the errors on the links between sensors. In this section, we discuss the process of link-error modeling. The error on the link from v_i to v_j is denoted p_{ij}. We use regression modeling between the time series of both streams to estimate the value of p_{ij}. If the time series at v_i and v_j are given by $x_1 \ldots x_t$ and $y_1 \ldots y_t$, respectively, then we can model $y_1 \ldots y_t$ as a function of $x_1 \ldots x_t$ with the following linear regression model for a window size of length w:

$$y_m = \sum_{n=1}^{w} a_n \cdot x_{m-w+n} \text{ for } w \leq m \leq t \tag{3.1}$$

Here, $a_1 \ldots a_w$ are *regression coefficients*, which are used to model the dependence of $y_1 \ldots y_t$ on $x_1 \ldots x_t$. The values of these coefficients are specific to the particular pair of nodes that are being tested. Historical samples from the underlying streams are used in conjunction with mean-square regression analysis to estimate the coefficients [1]. Once these regression coefficients have been estimated, they are used to estimate the error of prediction. Here, the *Recursive Least Square (RLS)* method [67] is used to estimate the coefficients. In this method, if there are n sample pairs consisting of vectors $\vec{x}_i$ of w independent values and dependent values y_i, then a *gain matrix* G_i and coefficient vector $\vec{a}_i$ are incrementally updated using $\vec{x}_i$, y_i, G_{i-1}, and $\vec{a}_{i-1}$, such that $\vec{a} = \vec{a}_n$. This method also allows for an optional *forget exponent*, which exponentially decreases the effect of samples on the coefficients with time. This can be done in order to make the approach dy-

namic, in terms of continuously updating the regression coefficients with changes in the underlying streams.

Once the coefficients have been estimated, they are used to compute an *estimated value* y'_m. The estimated values are compared to the true values in order to estimate the underlying error. The square error of prediction is given by $r_m^2 = (y_m - y'_m)^2$, and the root mean-square error of prediction over the entire window is $\sqrt{\frac{\sum_{m=w}^{t} {r_m}^2}{t-w}}$. This is the error-value p_{ij} which is assigned to the link between v_i and v_j. We note that the value of p_{ij} only needs to be computed between those vertices v_i and v_j for which a link e_{ij} is present in the information network.

3.3 Sensor Selection Algorithms

In this section, we design the algorithms for selection of sensor streams. First, we prove the hardness of the problem of stream selection by reductions from two well-known problems: the knapsack problem and the dominating set problem. The reduction from both problems highlights the fact that the problem is hard even in limited cases: when sensors cannot predict the streams of one another, and when sensors are able to predict the streams of one other with complete accuracy.

Theorem 1. The stream selection problem is NP-hard.

Proof. We prove the hardness by a reduction from the knapsack problem to the error reduction (maximization) problem. Consider a knapsack with capacity W and n items with weight $w(l_i)$ and value $v(l_i)$, for each item l_i. The objective is to determine if there is a subset of the items whose total weight does not exceed

W and whose total value is at least V. This can be modeled as a sensor selection problem in which there are only sensors and no edges (predictability relationships between sensors). Each sensor vertex v_i corresponds to a knapsack item l_i, with $c_i = w(l_i)$ and $z_i = v(l_i)$. The equivalent decision problem is to determine if there is a subset of sensors whose total cost does not exceed W and whose total error reduction is at least V.

Alternatively, we can prove NP-hardness by a reduction from the dominating set problem to the maximization version of sensor selection problem. In the dominating set problem, one is given a graph $G = (V,E)$, in which the objective is to determine if there is a set of vertices $V' \subseteq V$ of size k such that, for every vertex $v \in V$, either $v \in V'$, or there is an edge from some vertex in V' to v. This can be modeled as the sensor selection problem by using the same graph and assigning each vertex a cost and weight of 1 and each edge a weight of 1, such that the maximum error reduction is $|E|$. There is a one-to-one correspondence between the vertices selected in the solution to the sensor selection problem to those in the dominating set problem. The equivalent decision problem is to determine if there is a set of sensors with budget k whose error reduction equals $|E|$. □

Since the sensor selection problem is NP-hard, even for the limited cases described above, bounds on the approximability of the optimal solution remain to be shown. An upper bound on the approximation factor of any polynomial time algorithm is proven by a reduction from the *max-k cover* problem, which is defined as follows: Given a collection of sets $S = \{S_1, \ldots, S_m\}$ that is defined over a set of elements $\{x_1, \ldots, x_n\}$, choose k sets such that the cardinality of the union of these

sets is maximal. This is similar to *set cover*, except that in set cover, the objective is to find the smallest number of sets that cover all elements.

Theorem 2. No polynomial time algorithm for sensor selection can achieve an approximation factor better than $\frac{e-1}{e}$, unless $P = NP$.

Proof Sketch: This is derived by a reduction from the max-k cover problem to the maximized error reduction problem. We associate each set $S_i \in S$ with a vertex v_i with cost 1 and weight 0. We associate each element x_i with a vertex v_i with cost ∞ and weight 1. For each set S_i and each $x_j \in S_i$, we assign an edge e_{ij} between the corresponding vertices v_i and v_j with weight 1.

A solution to such an instance of the sensor selection problem will only select vertices associated with sets in S. Furthermore, only edges to vertices associated with elements covered by the sets associated with selected vertices contribute to the error reduction. Therefore, a solution to such an instance determines which sets maximize the weight of the covered elements within the budget constraints.

According to the reduction above, the budget and maximum error reduction in the sensor selection problem equals k and the maximum number of elements covered in the corresponding coverage problem. It was already proven that the best achievable approximation factor for the latter problem is $\frac{e-1}{e}$, unless $P = NP$ [16]. The result follows.

Note that the complement to the maximization problem, the error minimization problem, has no guaranteed approximation bounds, which can be proven by a simple reduction from the facilities location problem. For this reason, the focus here is on the approximability of the maximization problem.

The relationship of the sensor selection problem to other NP-hard problems is a useful exercise in cases where the objective functions show exact value correspondence, because it provides an idea of the kinds of algorithms which one may use in order to provide effective heuristic solutions. Fortunately, the link-based sensor selection problem has an interesting relationship with the *generalized maximum coverage problem* [13]. This relationship suggests the use of a carefully designed greedy mechanism [13], which defines and uses the concept of *residual density* to regulate the sensor selection process. We will show that such an approach provides an approximation factor of $\frac{e-1}{2 \cdot e-1}$ for sensor selection, where e is the base of the natural logarithm. This result applies to the maximization problem. The complement to the maximization problem, the error minimization problem, has no guaranteed approximation bounds, which can be proven by a simple reduction from the facilities location problem.

We propose two classes of solutions to the sensor selection problem. The first is integer programming, which can be used for either the minimization or maximization problems. We also present a greedy approximation algorithm for the maximization problem. We now describe both of these classes of solutions in detail.

3.3.1 Integer Programming Formulation

An integer programming formulation is a natural way of representing the problem. One advantage of using integer programming formulations is that they can be solved quite efficiently by a number of "off-the-shelf" tools. Furthermore, such

tools can also provide an idea of the quality of the solution, since they often use linear programming relaxations in order to provide an optimistic bound on solution quality.

The minimization problem can be solved by transforming it to an integer programming formulation that is analogous to that of the p-medians problem [49]. We use two sets of variables: one for describing the selection of sensors, and another for describing the predictability between different sensors. The set of variables y_i indicate whether or not the stream from sensor v_i is selected for collection. Here, y_i is a binary variable which takes on the value of 1 when the sensor v_i is selected for collection. The binary variable x_{ij} indicates whether or not the stream from sensor v_i is used to predict v_j. The variables x_{ij} are defined only for the cases when the edge e_{ij} is present in the network.

The objective function is to minimize the sum of the weight of the sensors which cannot be predicted at all (because the sensors are not selected and no neighboring sensors predict them), and the weighted prediction error of the edges x_{ij} that are used to make regression-based predictions of sensors that are not selected. The former is the first term in the objective function of the minimization integer program below, and the latter is the second term in the objective function. A number of constraints are defined in order to establish the budget constraints and the relationship between the sensor selection and edge-based regression. The first constraint ensures that the cost of the selected sensors is at most equal to a budget B. The second constraint ensures that a sensor can be predicted by either its direct selection or by prediction from *at most* one neighboring (inlinking) sensor.

The third constraint ensures that an edge may be used for prediction only if the source of the edge is included in the set of selected sensors. The final constraint ensures that all variables are integer.

$$\text{Min.} \sum_{v_i \in V} u_i \cdot q_i \cdot (1 - y_i - \sum_{j:e_{ji} \in E} x_{ji}) + \sum_{e_{ij} \in E} u_j \cdot p_{ij} \cdot x_{ij}$$

$$\text{subj. to:} \sum_{v_i \in V} c_i \cdot y_i \leq B, \qquad y_i + \sum_{j:e_{ji} \in E} x_{ji} \leq 1 \ \forall v_i \in V$$

$$x_{ij} \leq y_i, \quad y_i, x_{ij} \in \{0,1\} \qquad \forall v_i \in V, e_{ij} \in E$$

The maximization problem can be solved by a similar integer programming formulation, in which the objective is to maximize the total reduction in the error as a result of sensor selection. We use the same set of decision variables y_i and x_{ij}. The reduction in error as a result of selecting sensors can be of two types, which is reflected in the two kinds of the terms in the objective function below: **(1)** For the case of selected sensors, the reduction in error is equal to the maximum importance weighted error, which is also equal to $z_j = u_j \cdot q_j$. This is reflected in the first term in the objective function of the formulation below. **(2)** For the case of sensors which are not selected, but are predicted by a neighboring sensor, the reduction in error is equal to $u_j \cdot (q_j - p_{ij})$. This is reflected in the second term of the objective function below. Therefore, the decision problem may be formulated

as follows:

$$\text{Maximize} \sum_{v_i \in V} u_i \cdot q_i \cdot y_i + \sum_{e_{ij} \in E} u_j \cdot (q_j - p_{ij}) \cdot x_{ij}$$

$$\text{subj. to:} \sum_{v_i \in V} c_i \cdot y_i \leq B, \quad y_i + \sum_{j : e_{ji} \in E} x_{ji} \leq 1 \quad \forall v_i \in V$$

$$x_{ij} \leq y_i \quad y_i, x_{ij} \in \{0,1\} \qquad \forall v_i \in V, e_{ij} \in E$$

The constraints in the maximization and minimization formulations are similar, since they use the same variables and constraints. The main difference is in the formulation of the objective function. An advantage of formulating the problem as a maximization problem is that it lends itself to the use of other algorithms which can be implemented efficiently and also have a number of nice approximation properties. This also provides us with bounds on the quality of the solutions. We will discuss one such algorithm below, which leverages the use of the maximization formulation.

3.3.2 Greedy Approximation Algorithm

The aforementioned relationship of the maximization version of the sensor selection problem to the maximum coverage problem [13, 27] provides us with some ideas about the approximation algorithms that one may use to solve the problem. We will leverage this relationship to design an algorithm with a constant approximation factor.

In this algorithm, the sensor with the highest *residual density of importance-*

Algorithm *GreedySelect(SensorSet: V; EdgeSet: E;*
Budget: B; ImportanceVector: $\overline{u}$; CostVector: $\overline{c}$;
MaxSensorErrors: $\overline{q}$; RegressionErrorVector: $\overline{p_{ij}}$);
begin
$L = \{\}$; { Current Sensor Set }
repeat
Determine the sensor with largest value of
residual density $R(L, v_i)/c_i$, if one exists
such that $L \cup \{v_i\}$ is within budget B;
if such a v_i exists, then add to L;
until(no vertex v_i can be added to L);
Determine single vertex v_s with
largest value of $R(\{\}, v_s)$;
return the better of L and $\{v_s\}$
in terms of overall regression based error;
end

Figure 3.1: Greedy sensor selection

weighted error reduction among the not-yet-selected sensors is added to the set of selected sensors in each round if its cost does not violate the budget constraints. Therefore, we need to define the concept of *density of importance-weighted error reduction*. For a given set of sensors L that have already been selected, we can calculate the optimal assignments for regression-based prediction and use it to compute the importance-weighted error of prediction for this set of sensors. We can then calculate the further (or *residual*) importance-weighted reduction in error of prediction by adding sensor v_i to L, which we denote $R(L, v_i)$. In general, the value of $R(L, v_i)$ is non-increasing with increasing size of L. We note that this reduction in error is computed in terms of the regression errors p_{ij}, the sensor importance u_i, and the maximum sensor errors q_i. The residual density of a sensor

is the *residual error reduction* $R(L, v_i)$ divided by the cost c_i of sensor v_i. In other words, the residual density of sensor v_i *with respect to* the current sensor set L is given by $R(L, v_i)/c_i$.

It turns out that this approach may sometimes *not* have an approximation bound because of special cases in which a single sensor may provide excellent prediction of all other sensors, yet has a high cost. Consider the following example [27]: Let $V = \{v_1, v_2\}$, such that $z_1 = 1, z_2 = p, c_1 = 1, c_2 = (p+1)$. The residual density of v_1 and v_2 in the first round is 1 and $p/(p+1)$, respectively. If the budget is p, then only v_1 will be selected with a weight of 1, while the optimal solution is v_2 with a weight of p. The approximation factor is p and is therefore unbounded. In order to neutralize the effect of such cases, we simply consider a solution in which we pick the *single* vertex v_i (within the budget constraints) that has the largest value of $R(\{\}, v_i)$. Note that we are not dividing by the cost c_i in this case. Since the coverage to cost ratio may be less than that of other subsets, greedy selection does not necessarily select this subset. The *best set* among the two possibilities (greedy selection and the singleton set) is reported as the optimal solution. The overall algorithm for sensor selection is illustrated in Figure 3.1.

A key step in this algorithm is the computation of the residual error reduction $R(L, v_i)$ in each step. We note that each sensor v_j is either selected in the set L, not predicted at all, or is predicted by the vertex $v_k \in L$ with the smallest value of p_{kj} and a corresponding importance weighted error of $u_j \cdot p_{kj}$. If the sensor v_j is currently predicted by v_k, and v_i has lower error p_{ij} of regression-based predictability than the current value p_{kj}, then this reduces the error by $u_j \cdot (p_{kj} -$

p_{ij}). On the other hand, if sensor v_j is currently not predicted at all by any sensor, then the error is reduced by $u_j \cdot (q_j - p_{ij})$. Otherwise, there is no error reduction of sensor v_j. Therefore, the residual error reduction is defined as the sum of these values over all the sensors. The residual error reduction density is then determined by dividing this value by the cost c_i. It remains to show that the above algorithm has an approximation factor of $\frac{e-1}{2 \cdot e-1}$

Theorem 3. The greedy algorithm has a constant approximation factor of $\frac{e-1}{2 \cdot e-1}$ of the optimal solution, where e is the base of the natural logarithm.

Proof Sketch: The proof of the approximation bound may be derived by modeling the sensor selection problem as a *generalized maximum coverage problem* [13]. In the generalized maximum coverage problem, there is a set of elements $\mathscr{E}$ (where the i^{th} element is denoted by l_i) and a set of bins $\mathscr{B}$ (where the i^{th} bin is denoted by b_i). Each element is assigned a unique positive profit and non-negative weight for assignment to each bin. Additionally, each bin is assigned a weight as overhead for using the bin. The objective is to find a selection of bins and an assignment of elements to those bins with maximum profit and whose total weight is within some budget constraint B. The sensor selection problem can be modeled as an instance of the generalized coverage problem as follows. We associate each vertex v_i with a bin b_i and an element l_i, and set the weight of each b_i to c_i. If the edge e_{ij} exists or if $i = j$, it is possible to use sensor v_i to predict v_j. In this case, we set the weight of assigning l_j to b_i to 0 and set the corresponding profit to $u_j \cdot q_j$ when $i = j$ and $u_j \cdot (q_j - p_{ij})$ when $i \neq j$. Otherwise, since e_{ij} does not exist, then v_i can not predict v_j, and so we set the weight of assigning l_j to b_i to ∞

to prevent such an assignment.

It can be shown that there is a one-to-one correspondence between solutions to instances of the two problems with identical objective function values. Furthermore, application of the greedy algorithm for the generalized maximum coverage problem [13] to the sensor-based instance of the maximum-coverage problem results in the same sequence of steps as proposed by the greedy sensor selection scheme. Therefore, the approximation factor of the sensor selection scheme is same as that of the greedy algorithm in [13], which is $\frac{e-1}{2 \cdot e-1}$.

Note that is sufficient to select only the best singleton set in addition to the greedy solution to guarantee this approximation bound, although adding sensors will further improve the solution. If desired, the approximation bound can be further improved to $\frac{e-1}{e}$ by adapting a partial enumeration-based method [13, 33] to the sensor selection problem. Although this improves the worst-case bound, partial enumeration is time-consuming in practice and makes the approach less efficient. Furthermore, our experimental results show that our earlier approach can obtain nearly optimal results most of the time, as is evidenced by the comparison of our results with that of an integer programming solver. Therefore, we retain our afore-mentioned (simpler) approach with approximation bound of $\frac{e-1}{2 \cdot e-1}$ in order to obtain more practical and efficient results.

3.4 Experimental Results

The goal of the testing was to show that the use of virtual information network links is useful for performing sensor selection and also results in great efficiency improvements. The data sets needed some further preparation in order to create the virtual links. While these data sets contain multiple time series, they do not contain virtual links as in an information network. Fortunately, we can use the meta-information associated with these data sets in order to create virtual links.

3.4.1 Data Sets

We divided each stream into a *training set* and a *test set*. The training set was used to derive the regression coefficients. The coefficients were then used to calculate the prediction error on test data. The data sets used are as follows:

Intel Berkeley Lab Data

The Intel Berkeley Lab data set [33] contains temperature, humidity, and light data collected from 54 motes placed in the Intel Berkeley Research lab. Each reading is assigned an *epoch number* that corresponds to the time at which the reading was taken. Only data up to epoch 60,000 is available. We divided this data set in half, using the first 30,000 epochs for training data and the remaining epochs for test data. We used the distance between the motes in order to generate the information network links that were used for predictions. That is, two motes were linked if the distance between them was lower than a given threshold. Links were allowed

only between sensors of the same type. The logic in allowing such links was that they provided domain knowledge about the actual relationships between different sensors, which were also useful for the purposes of predictability.

EPANET Water Data

We used EPANET 2.0 to simulate water flow in a water pipe network with 126 junctions between various pipes [47]. This topology provides useful information about the relationship between chlorine concentrations at the junctions and was used in order to generate the information network links for predictive purposes. We sampled the chlorine levels at these junctions every 2 minutes over 480 hours, in which the demand at each junction changes every 30 minutes and follows a cyclic pattern that repeats itself every 48 hours.

Further Data Preparation

We normalized the values in each stream to the number of standard deviations from the mean and set the maximum error of prediction for each vertex to 10, the difference between 5 standard deviations above or below the mean. This is essentially equal to the maximum error q_i of each sensor. In addition, we needed to assign costs and importance values to the vertices. All importance values u_i for the sensors were set to 1, whereas costs differed between two scenarios **(a)** In the first scenario, all costs were uniform. **(b)** In the second scenario, the costs were designed to be non-uniform, and made to vary from a Zipf distribution. Thus, the sensors were randomly ordered and assigned a corresponding index i. The cost of

the i^{th} sensor was then assumed to be $1/(i+1)^{\theta}$, $\forall 1 \leq i \leq n$.

3.4.2 Baseline Algorithms

We also used a sampling strategy as a baseline in order to test the effectiveness of the algorithm. The sampling strategy is to select vertices from V one by one at random and add them to S, on condition that they do not violate the budget constraints, until there are no more vertices to add. In order to further improve the effectiveness of sampling, we performed the random selection multiple times and picked the best of these selections. For the purpose of this paper, we used fifty samples.

3.4.3 Evaluation Measures

The stream selection mechanism was tested for effectiveness, efficiency, and sensitivity. In each case, we tested both the complete graph of links as well as the information network graph in order to show the efficiency advantages of encoding domain knowledge in the selection process. We used the selected sensor streams (and corresponding prediction assignments) in order to perform actual stream-to-stream regression and thereby predict the non-selected streams. The error of actual prediction is reported over different budgets. To evaluate efficiency, we determined the time required for sensor selection with the use of different techniques. We compare the running time over different budgets. We show that the use of carefully selected information network links can be useful in greatly im-

proving the efficiency of the selection technique. We also tested the sensitivity of the technique to varying costs.

3.4.4 Effectiveness and Efficiency Results

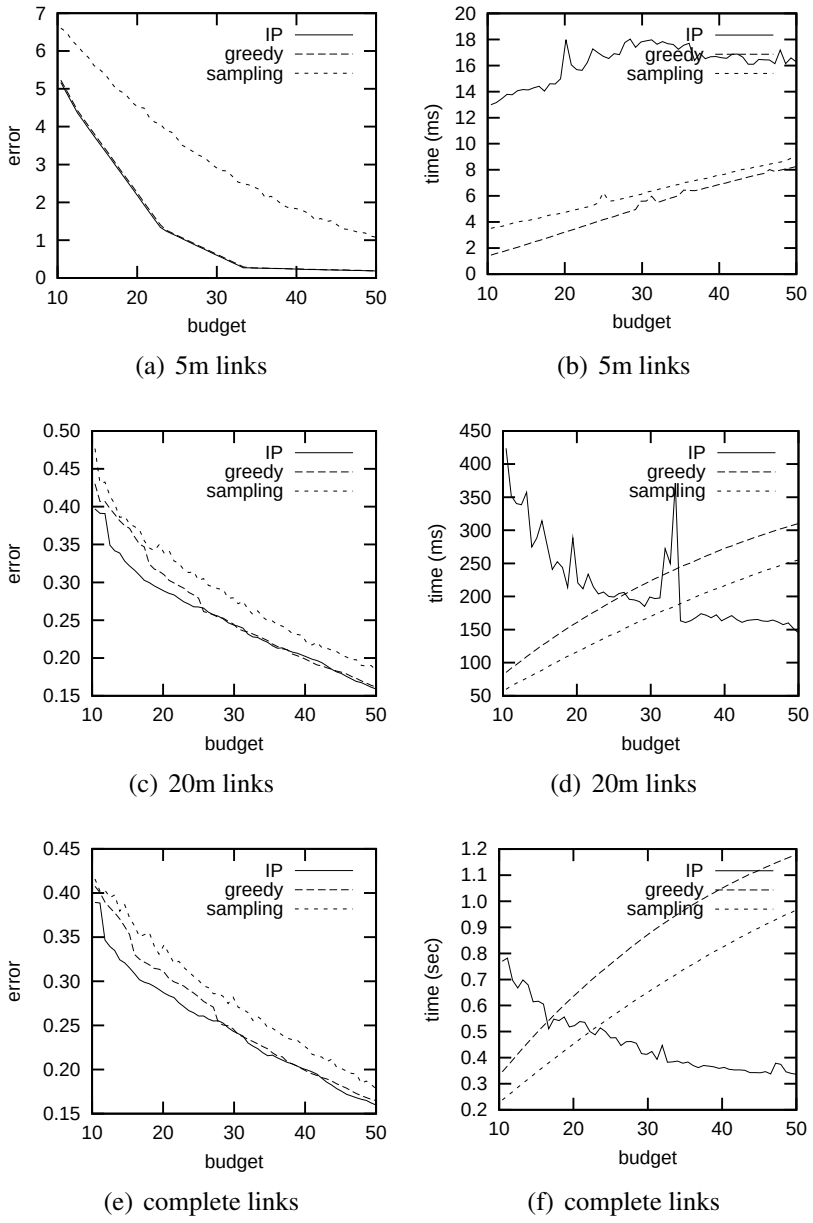
We used the integer programming, greedy, and sampling methods to derive sensor selections with the use of virtual links. These were utilized in order to compute the optimal selection, which was then run over the test data to get the regression-based prediction errors. The IP solver was set to search for an integer optimal solution or terminate after ninety seconds. In the case when the solver terminated, the best solution found so far was used, which is therefore an approximation to the optimal solution. In only a few cases did the solver timeout. In each case, the actual error prediction is computed and reported. We first compare how the error and computation time of all three methods vary with the budget, ranging from 10% to 50% of the total cost of all vertices. We then examine the sensitivity of the method to the skew in the costs across different sensors.

Intel Berkeley Lab Data

Unless otherwise mentioned, the default window size was set to 8. We present in which the temperature, humidity, and light streams were combined into one logical information network structure, thus tripling the number of streams. Virtual links were allowed between like streams only. In the simulations, we established virtual links according to the distance between sensors. In the following figures, we compare results for using links between sensors that are within five meters

of each other ("5m links"), twenty meters ("20m links"), and links between all sensors ("complete links"). Figure 3.2 presents the effectiveness and efficiency results for the Intel Berkeley data over different link types. In this case, uniform costs were used. In Figure 3.2(a), we present the error rates with increasing budget for the case in which links were placed between sensors only when they were at a distance of 5 meters or less. On the X-axis, we present the budget as a percentage of the total sensor cost, and on the Y-axis we present the error of regression modeling. It is evident that the error decreases as the budget is increased and more sensors are available to perform the prediction. Another observation is that the curve for the greedy strategy overlaps that of the IP strategy. Since the IP strategy often terminated with an optimal solution, it follows that the greedy strategy was close to optimal as well in practice. Furthermore, both strategies are significantly superior to the sampling strategy over the entire range of values for the budget.

The efficiency results for the Intel Berkeley data are presented in 3.2(b). The budget is presented in the X-axis, and the running times are illustrated on the Y-axis. It is evident that the greedy algorithm was extremely efficient and provided lower execution times than the other methods. This trend was true in the case of all the different linkage-based representations. The greater efficiency of the greedy method is in spite of the fact that the greedy technique provided almost optimal results in most cases. This suggests that the greedy method provides the best tradeoff between effectiveness and efficiency. In addition, both the greedy and the sampling method had running times which increased gradually with the budget. This was because of the natural iterative way in which the nodes were



(a) 5m links (b) 5m links

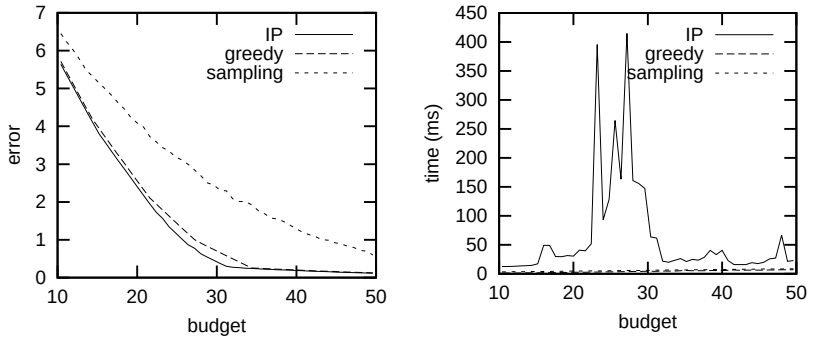
(c) 20m links (d) 20m links

(e) complete links (f) complete links

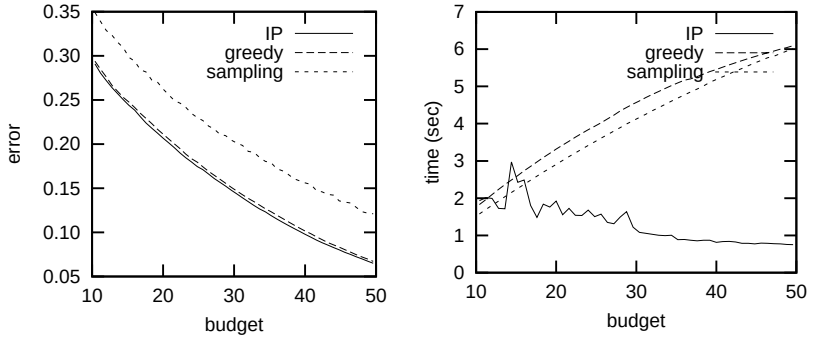
Figure 3.2: Errors and execution times for Intel Berkeley data

added to the solution in the case of the greedy and sampling strategies. In the case of the IP solution, the running times were much more erratic, because the running time of the solver was dependent upon the time which it took to arrive at a near-optimal solution. This often depended on the starting point and many other factors that were more important than the size of the problem. In fact, the IP solver often required less time for larger budgets, because the optimal solution was much easier for the solver to find in such cases. For larger budgets, the IP solver sometimes performed more efficiently than the greedy algorithm. However, since the resource constrained scenarios are the more challenging and interesting case, the greedy solution turned out to be the most valuable on an overall basis.

Analogous results for the case when the links were defined by 20m (or less) distances and the complete set of links are illustrated in Figures 3.2(c)–(f). It is evident that the use of a larger number of links encodes a greater amount of information, and therefore reduces the error. However, such an error reduction comes at a great cost, as the execution times increase tremendously as the number of links are increased. For example, the use of 20m links is almost 20 times slower than using 5m links, whereas the use of the complete graph is two orders of magnitude slower than using 5m links. Since the sensor selection problem often may need to be repeatedly applied on changing cost and importance scenario, the efficiency of the method is paramount to using the method successfully in a variety of scenarios.



(a) junction topology graph (b) junction topology graph



(c) complete graph (d) complete graph

Figure 3.3: Effectiveness and efficiency results for EPANET data

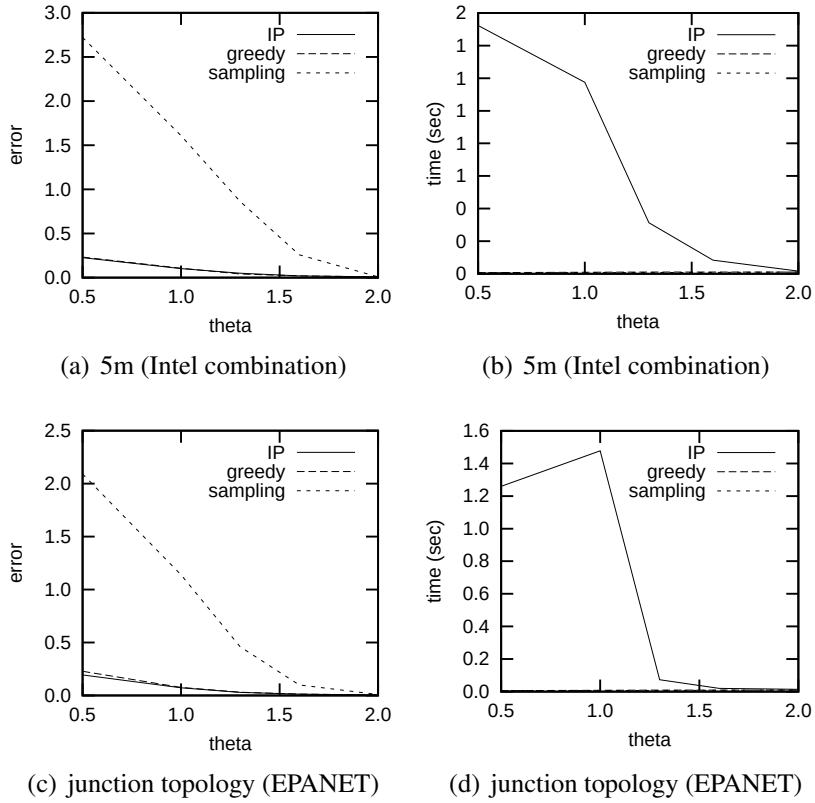
EPANET Water Data

Since the pipes are actual connections between junctions, their presence has predictability power in terms of the chlorine concentrations measured by sensors. Therefore, we used the topology of the pipe junctions as our underlying network. In addition, we also tested using the predictability power of all pairs of junctions (which we refer to as the "complete graph"). Figure 3.3 presents the results for both cases with uniform costs. The effectiveness and efficiency results for the case where the pipe topology is used are presented in Figures 3.3(a) and 3.3(b),

respectively. The cases in which the complete graph is used are presented in Figures 3.3(c) and 3.3(d), respectively. We make two main observations in this case. The first is that the prediction error of the greedy strategy matched the error of the IP strategy almost exactly in both scenarios. In particular, the baseline sampling strategy performed quite poorly in this case. This is another demonstration of the fact that the greedy strategy not only has an approximation bound, but is also quite effective in practice. The second is that the IP strategy is much more erratic in this case as compared to the Intel data. For the case in which the link topology was used, we found that the running time for the IP solver peaked suddenly at intermediate values of the budget. This is because intermediate values of the budget provided the most realistic combinations to the IP solver. As a result, the running times could also be greater in these cases. Because of the erratic nature of the IP solver in terms of efficiency, it is evident that the greedy approach is much more desirable, especially in cases in which the importance of the sensors varies over time and the solution needs to be repeatedly recomputed.

3.4.5 Cost Sensitivity Analysis

We also tested the case in which the costs of the sensors are non-uniform and drawn from a Zipf distribution. The cost of the i^{th} sensor was assumed to be $1/(i+1)^{\theta}$, $\forall 1 \leq i \leq n$. The value of θ varied in the range $[0.5, 2.0]$. The budget was fixed at 30% of the total cost of all sensors. The results are presented in Figure 3.4. In each case, the Zipf parameter is represented on the X-axis. The errors and execution times for the Intel combination data are presented in Figures 3.4(a) and



(a) 5m (Intel combination) (b) 5m (Intel combination)

(c) junction topology (EPANET) (d) junction topology (EPANET)

Figure 3.4: Sensitivity with cost skew (budget fixed at 30%)

3.4(b), respectively, whereas that for the EPANET data set are presented in Figures 3.4(c) and 3.4(d), respectively. One trend that is evident from the different figures is that the error differences between the sampling and the other two strategies are greater for lower levels of skew. This is because when the skew level is very high, the effectiveness is greatly dominated in the inclusions of one or two sensors with very high costs. Since the sampling strategy picks the best set over many iterations, it was able to pick out those few sensors in those cases. Nevertheless, the greedy strategy turned out to be quite robust in this case, in terms of both

execution and running times. The IP strategy again exhibited its unpredictable behavior in terms of the running times, even in terms of the variation over different values of the skew parameter θ. This again suggests that certain instances of the problem can be more difficult for the solver, a result of which they require larger running times.

3.5 Related work

It was already shown that dominating set can be reduced to sensor selection. The problem defined in the reduction is actually maximum dominating set, in which the objective is to choose the k vertices that dominate the most vertices in the graph. The equivalent set cover problem is known as the maximum k-cover problem, as described above. Feige [16] proved that no polynomial time algorithm for maximum k-cover can guarantee an approximation factor greater than $1-1/e$, unless $P=NP$. Khuller et al. [27] independently proved a weaker result, that no approximation factor greater than $1-1/e$ exists unless $NP \subseteq DPTIME(n^{\log\log n})$. The essence of the proof is that if there is a better approximation, then set cover can be solved by a factor of $(1-\varepsilon)\log n$ for some $\varepsilon > 0$, which Feige [16] previously proved is not possible unless $NP \subseteq DPTIME(n^{\log\log n})$.

If the vertex costs equal one and the edge weights equal the end vertex weights, but the vertex weights are still allowed to vary, then this is equivalent to the *maximal covering location problem* [12]. The objective in this problem is to select p facility locations that service the largest population within a fixed service dis-

tance of each facility. Church and ReVelle [12] define a greedy selection algorithm in which locations are selected one by one according to those which service the greatest population not serviced by any other location selected so far. They describe an enhancement to the greedy algorithm by replacing already selected facility locations by non-selected facilities that service a greater population, and evaluate the performance of their solutions by upper bounding optimal results with an integer programming solution. For trees, Megiddo et al. [44], present an $O(n^2p)$ solution, while McHugh and Perl [42] provide a $O(np^2)$ dynamic programming solution. Hochbaum and Pathria [24] prove a tight bound of $1-1/e$ on the approximation factor of the greedy algorithm for the equivalent set cover problem. Given the proofs of Khuller et al. [27] and Feige [16] that the best possible approximation factor of max k-cover is $1-1/e$, a tightness proof would not have been necessary in this case.

The *constrained maximum weight domination problem* [42] generalizes the problem in which locations are assigned variable costs and the objective is to select a set of locations within a budget constraint B that maximizes coverage. For trees, McHugh and Perl [42] provide a $O(nB^2)$ pseudo-polynomial dynamic programming solution, and Megiddo et al. [44] claim that their algorithm can be modified to solve this problem in pseudo-polynomial time as well. Khuller et al. [27], define and analyze the greedy algorithm for the equivalent set cover problem, which they call the *budgeted maximum coverage problem*. The greedy algorithm of Hochbaum and Pathria [24] is modified to select the set within the budget constraints that has the greatest *residual density* each round. They prove

that the modified greedy algorithm has a $1-1/\sqrt{(e)}$ approximation factor, but could not show that this is tight. They prove that greedy selection with partial enumeration has an approximation factor of $1-1/e$, which is tight given their result mentioned above for max k-cover.

The *generalized maximum coverage problem* [13] best represents the fractional dominating set problem. The objective is to assign a set of elements to a set of bins within a budget constraint B with maximum profit, in which a cost and profit are associated with each assignment of an element to a bin, and cost is associated with each bin to which elements are assigned. A greedy selection algorithm similar to the one presented by Khuller et al. [27], for the budgeted maximum coverage problem has an approximation factor of $\frac{e-1}{2e-1}-\varepsilon$. In this algorithm, the bin, assignment of new elements to the bin, and reassignment of old elements to the new bin with the largest residual density is selected each round. By using a method of partial enumeration, the approximation factor is improved to $\frac{e-1}{e}-\varepsilon$.

Sviridenko [58] generalizes the budgeted maximum coverage problem differently by assuming the coverage by a collection of sets S is defined by a function $f(S)$, as described in the generalized problem definition above. If f is submodular, then greedy selection with partial enumeration has a $\frac{e-1}{e}$ approximation factor as well. Note that choosing the edge with the maximum weight in the fractional dominating set problem is a submodular function, so this bound applies to the simple definition of the problem as well.

Chapter 4

Increasing Threshold Search

Increasing threshold search is a type of search with iteratively increasing search extents. It is specifically intended to find any agent in a multi-agent system (MAS) associated with the *best* value (lowest or highest, depending on the application) amongst a set of agents associated with a value to the searcher. The variable search extent is the range of acceptable values that replying agents need to comply with. It is applicable in environments in which: (a) each agent is associated with a single value, which may be an intrinsic value associated with the agent or a mapping from a combination of values associated with the agent; and (b) the searcher can publish a request to all other agents.

This chapter thoroughly analyzes the problem of deriving the optimal threshold-based search sequence in settings similar to those above. It is assumed that all agent values are associated with a common distribution, which is known to the searching agent and remains constant over time [2, 7, 43]. Learning the actual

value of an agent incurs some cost to the searcher, the searched agent, or both (e.g., consuming some of the agents's resources for communicating with each other, querying for relevant information, and processing the information received). Such costs are commonly incurred in multi-agent systems in the absence of a central source that can supply full, immediate, and reliable information on the environment and the state of the other agents that can be found [2, 11, 26]. Additionally, there are possible publishing costs, which remain constant throughout the search process. The optimal sequence should trade off the expected increase in the number of search rounds with the expected decrease in the number of replying agents.

In the next two sections, we formally introduce and analyze the optimal increasing threshold search for the class of environments described above. In Section 4.3, we illustrate the properties of the optimal strategy with different values of the problem parameters. The benefits of using the optimal strategy are illustrated by comparing it to adaptations of three well studied expanding ring strategies [3, 9, 22] to the problem considered in this paper. In Section 4.4, we extend the analysis to the case where a group of the best-valued agents needs to be found; for example, when several readings are required by the application. We show that a similar method of analysis can be applied to this case. In Section 4.5, we show how increasing threshold search is applicable to *economic search* [2, 40, 43], in which the searcher is not necessarily constrained to finding the best-valued agent, but rather attempts to optimize a function that integrates both search costs and the value of the agent ultimately found. We also show how economic search

strategies can be combined with threshold-based searches to further reduce overall costs. Throughout the chapter, we illustrate the properties of the various search techniques using data from a synthetic environment.

4.1 Model Formulation

We consider an agent searching in an environment where N other agents, applicable to its search, can be found. (Appendix C contains the complete set of notations used throughout the paper.) Each of the N agents is characterized by its value to the searcher. As in most search-related models, the values are assumed to be associated with a common continuous distribution described by a PDF $f(x)$ and a CDF $F(x)$, defined over the interval $[x_{min}, x_{max}]$ [7, 10, 43, 54]. The searcher agent is assumed to be ignorant of the value associated with each of the N agents, but acquainted with the overall distribution of values. The searcher is interested in finding the agent associated with the *best* value, which, depending on the application, is either the minimum or the maximum value. Without loss of generality, we assume that the best-valued agent is the one associated with the minimum value.

Obtaining the actual value of an agent incurs some cost. In its most general form, the cost of simultaneously obtaining the values of j other agents is $\beta(j)$ (where $\beta(0) = 0$ and $\beta(j)$ is strictly increasing in j) [4, 17, 45]. In order for the searcher to refine the population of agents whose values it obtains, it can publish a maximum threshold r on the agents' values, denoted a *reservation value*, requesting to communicate only with agents that comply with that threshold. If at

least one agent complies with r and communicates its value with the searcher, the search process terminates. Otherwise, the searcher sets a new reservation value $r' > r$ and repeats the process. This continues until at least one agent replies, out of which the agent associated with the minimum value is chosen. A strategy S is therefore a sequence $[r_1, r_2, \ldots]$ ($x_{min} < r_i < r_{i+1} \leq x_{max}, \forall i \geq 1$), where r_i denotes the reservation value to be used in the i^{th} search round. No constraints are placed on the number of rounds and, consequently, the length of the sequence. In order to guarantee search completeness when S is a finite sequence, the last reservation value in the sequence should equal x_{max}.

The process of initiating a search round and publishing the next reservation value is also associated with a cost α whose value is fixed (e.g., the cost of issuing a new call for bids or the cost of broadcasting a message). Note that this cost may actually be a function of N, but since N remains constant during the search, we can also consider this cost to be constant. The overall cost of a search round i is thus $\alpha + \beta(j)$, where j is the number of agents that comply with r_i. The expected accumulated cost of finding the best-valued agent when using strategy S is denoted $V(S)$. The searcher's goal is therefore to derive a strategy S^* that minimizes $V(S)$.

4.2 Analysis

Consider a searcher agent using a strategy $S = [r_1, \ldots, r_M = x_{max}]$. If the agent has to start the i^{th} search round, then there is necessarily no agent whose value is below r_{i-1}. The *a priori* probability of such a scenario is $(1 - F(r_{i-1}))^N$. Furthermore,

upon reaching the i^{th} round, the searcher agent can update its beliefs concerning the PDF and CDF of the values of the N agents, as it knows that these are necessarily in the interval $(r_{i-1},x_{max}]$. The PDF of the agents' values after publishing r_{i-1}, denoted $f(x|r_{i-1})$ $(0 < i \leq M)$, can thus be calculated as $(x_{min} \leq x \leq x_{max})$:

$$f(x|r_{i-1}) = \begin{cases} \frac{f(x)}{1-F(r_{i-1})} & i > 1 \wedge x > r_{i-1} \\ 0 & i > 1 \wedge x \leq r_{i-1} \\ f(x) & i = 1 \end{cases} \tag{4.1}$$

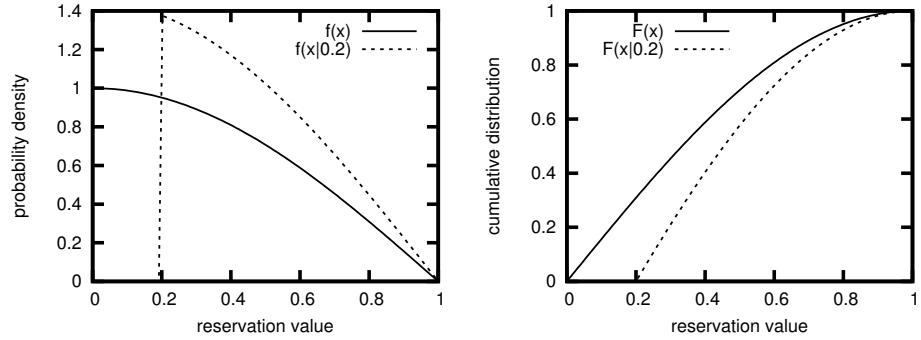
Similarly, the CDF of any of the agents' values after publishing r_{i-1}, denoted $F(x|r_{i-1})$ $(0 < i \leq M)$, can be calculated as $(x_{min} \leq x \leq x_{max})$:

$$F(x|r_{i-1}) = \begin{cases} \frac{F(x)-F(r_{i-1})}{1-F(r_{i-1})} & i > 1 \wedge x > r_{i-1} \\ 0 & i > 1 \wedge x < r_{i-1} \\ F(x) & i = 1 \end{cases} \tag{4.2}$$

See Figure 4.1 for an example PDF and CDF. The expected cost of the i^{th} round is thus:

$$\alpha + \sum_{j=1}^{N} \beta(j) \binom{N}{j} F(r_i|r_{i-1})^j (1 - F(r_i|r_{i-1}))^{N-j} \tag{4.3}$$

as it takes into account the cost of initiating the new search round and the expected cost of obtaining any possible number of agent values j $(0 < j \leq N)$ in round i. This latter cost is the sum, for all j, of the cost $\beta(j)$ of obtaining j agent values multiplied by all $\binom{N}{j}$ combinations of j agents whose values can be obtained,



(a) example PDF (b) example CDF

Figure 4.1: Example distribution

multiplied by the probability $F(r_i|r_{i-1})^j(1-F(r_i|r_{i-1}))^{N-j}$ of obtaining exactly j agent values in round i if none were found in previous rounds.

The probability of starting round i is $(1-F(r_{i-1}))^N$, which is the probability that no agent values are below the previous reservation value r_{i-1}. The expected cost of using strategy S is thus the sum of the expected cost of each of the M search rounds weighted by the probability of reaching that round:

$$V(S) = \sum_{i=1}^{M} \left(\alpha + \sum_{j=1}^{N} \beta(j) \binom{N}{j} F(r_i|r_{i-1})^j (1-F(r_i|r_{i-1}))^{N-j} \right) (1-F(r_{i-1}))^N \tag{4.4}$$

The probability of starting the i^{th} search round can alternatively be formulated as the probability that no values were obtained in each of the $i-1$ previous rounds,

expressed as $\prod_{j=1}^{i-1}(1-F(r_j|r_{j-1}))^N$. Therefore, (4.4) transforms into:

$$V(S) = \sum_{i=1}^{M}\left(\alpha + \sum_{j=1}^{N}\beta(j)\binom{N}{j}F(r_i|r_{i-1})^j(1-F(r_i|r_{i-1}))^{N-j}\right) \cdot \prod_{j=1}^{i-1}(1-F(r_j|r_{j-1}))^N \quad (4.5)$$

Note that the probability that the search terminates in round i can be calculated as the probability that all agent values are greater than r_{i-1}, $(1-F(r_{i-1}))^N$, minus the probability that all agent values are greater than r_i, $(1-F(r_i))^N$. Thus, the expected number of search rounds is

$$\sum_{i=1}^{M} i((1-F(r_{i-1}))^N - (1-F(r_i))^N) = \sum_{i=1}^{M}(1-F(r_{i-1}))^N - M(1-F(r_M))^N = \sum_{i=1}^{M}(1-F(r_{i-1}))^N \quad (4.6)$$

in which $F(r_0)=0$ and, by definition, $F(r_M)=F(r_{max})=1$. Accordingly, (4.4) can be understood differently by factoring the term $(1-F(r_{i-1}))^N$ into the remainder of the expression, resulting in the equation

$$V(S) = \alpha\sum_{i=1}^{M}(1-F(r_{i-1}))^N + \sum_{j=1}^{N}\beta(j)\binom{N}{j}\sum_{i=1}^{M}(F(r_i)-F(r_{i-1}))^j(1-F(r_i))^{N-j} \quad (4.7)$$

Here, the first term is the expected contribution of the fixed cost of publishing the threshold to the total search cost, which is α times the expected number of rounds. The second term is the expected cost of obtaining the values of all agents that

comply with the last published reservation value. This is calculated by summing the cost of obtaining j values, over $1 \leq j \leq N$, times the probability that j and only j agents comply with the reservation value published in the last search round.

For the specific case in which the reservation values are chosen from a finite set of L values $\{x_1, \ldots, x_L\}$, $x_{min} < x_i < x_{i+1} < x_L = x_{max}$ for all i, the optimal strategy can be derived with the following dynamic programming formulation in $O(L^2N)$ time:

$$C(L) = 0$$

$$C(l) = \min_{l+1 \leq i \leq L} \left\{ \alpha + \sum_{j=1}^{N} \beta(j) \binom{N}{j} F(x_i|x_l)^j (1 - F(x_i|x_l))^{N-j} + C(i)(1 - F(x_i|x_l))^N \right\}$$
$$\forall 0 \leq l < L \tag{4.8}$$

where $C(i)$ is the cost of continuing the search if a search up to value x_i failed to obtain any applicable values. The expected cost of the entire search is determined by $C(0)$.

For the general case in which the interval $[x_{min}, x_{max}]$ is continuous, the optimal search strategy must be derived using a different methodology since, as we prove in Theorem 4, the optimal search sequence is either a single search round in which the values of all agents are obtained or an infinite sequence of reservation values.

Theorem 4. The optimal sequence of reservation values is either $[r_1 = x_{max}]$ or the infinite sequence $[r_1, r_2, \ldots]$, $x_{min} < r_i < x_{max}, \forall i > 0$, where
$F(r_i|r_{i-1}) = F(r_j|r_{j-1}) = P$, for some P and $\forall i, j > 0$.

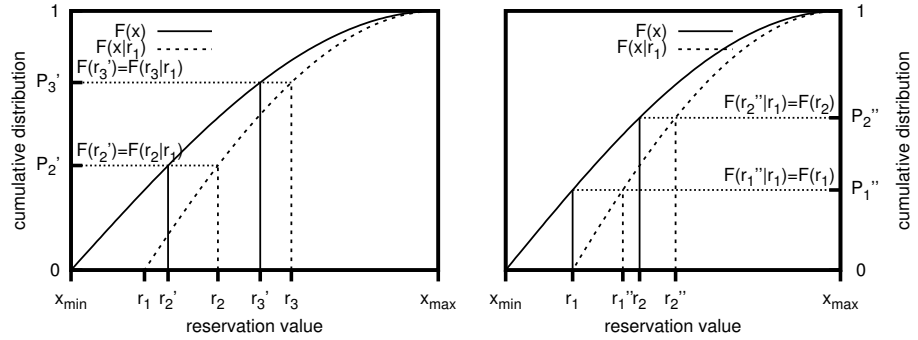
Proof. Assume the finite sequence $S_1 = [r_1, \ldots, r_M]$ is the optimal strategy. We use $S_2 = [r_2, \ldots, r_M]$ to denote the optimal strategy to be used if no agent is found in the first search round and denote its expected cost from that point on by $V^{(r_1)}(S_2)$. Using S_2, we construct an alternative strategy $S_1' = [r_2', \ldots, r_M']$ to be applied from the first round, where $F(r_i') = F(r_i|r_1) = P_i'$, $\forall 1 < i \leq M$ and for some P_i' (Figure 4.2(a)). The new strategy S_1' has an expected cost $V(S_1')$. Note that

$$F(r_i'|r_{i-1}') = \frac{F(r_i|r_1) - F(r_{i-1}|r_1)}{1 - F(r_{i-1}|r_1)} = \frac{\frac{F(r_i)-F(r_1)}{1-F(r_1)} - \frac{F(r_{i-1})-F(r_1)}{1-F(r_1)}}{\frac{1-F(r_{i-1})}{1-F(r_1)}}$$
$$= \frac{F(r_i) - F(r_{i-1})}{1 - F(r_{i-1})} = F(r_i|r_{i-1}) \tag{4.9}$$

By substituting $F(r_i|r_{i-1})$ with $F(r_i'|r_{i-1}')$ in (4.5), we obtain $V(S_1') = V^{(r_1)}(S_2)$. Since S_1 is the optimal strategy, $V(S_1) \leq V(S_1')$.

Now consider a new strategy $S_2' = [r_1'', \ldots, r_M'']$ to be applied from the second round on, where $F(r_i''|r_1) = F(r_i) = P_i''$, $\forall 1 \leq i \leq M$ (Figure 4.2(b)). We denote the expected cost of S_2' from that round on by $V^{(r_1)}(S_2')$. Note that $F(r_i''|r_{i-1}'') = F(r_i|r_{i-1}) = P_i''$, as above. According to (4.5), we obtain $V^{(r_1)}(S_2') = V(S_1)$. Since S_2 is the optimal strategy from the second round on, then $V^{(r_1)}(S_2) \leq V^{(r_1)}(S_2')$, resulting in $V(S_1) \leq V(S_1') = V^{(r_1)}(S_2) \leq V^{(r_1)}(S_2') = V(S_1)$, which can hold only if $V^{(r_1)}(S_2) = V(S_1)$.

The same logic can be applied to any search round $j \leq M$, resulting in $V^{(r_{j-1})}(S_j) = V(S_1)$. In particular, $V(S_1) = V^{(r_{M-1})}(S_M)$. However, we necessarily find all agents in the last (M^{th}) round, since $r_M = x_{max}$; thus, $V(S_M) = \alpha + \beta(N)$.



(a) Mapping from S_2 to S_1' (b) Mapping from S_1 to S_2'

Figure 4.2: Alternative strategies

Therefore, we obtain $V(S_1) = \alpha + \beta(N)$, which is equivalent to $S_1 = [r_1 = x_{max}]$. Any optimal strategy that has an expected cost less than $\alpha + \beta(N)$ must therefore consist of an infinite sequence and satisfy $F(r_i|r_{i-1}) = F(r_{i+1}|r_i) = P$, $\forall i > 0$ and some P. □

The immediate implication of Theorem 4 is that the optimal search strategy can be expressed as a single value $0 < P \leq 1$, denoted the *reservation probability*. This is the key to deriving the optimal sequence of reservation values. As outlined below, the searcher only needs to calculate P and then set each reservation value r_i such that $F(r_i|r_{i-1}) = P$. An important result of the ensuing analysis is that P is distribution independent. Consequently, only the actual reservation values need to be recalculated if belief about the distribution changes with new information. Additionally, it is possible to make distribution-independent observations given the remaining parameters, such as the expected cost of the optimal strategy and expected number of rounds.

First we show how to derive the reservation probability P. Since the optimal sequence is infinite and the expected cost from each round onwards is stationary, the expected cost of using P can be expressed with the following equation:

$$V(P) = \alpha + \sum_{j=1}^{N} \beta(j) \binom{N}{j} P^j (1-P)^{N-j} + (1-P)^N V(P) \tag{4.10}$$

Here, the first term is the fixed cost per round, the second term is the expected cost of obtaining any values, and the last term is the expected cost of continuing the search if necessary. Consequently:

$$V(P) = \frac{\alpha + \sum_{j=1}^{N} \beta(j) \binom{N}{j} P^j (1-P)^{N-j}}{1-(1-P)^N} \tag{4.11}$$

Just as in (4.7), (4.11) can be decomposed into two parts: $\frac{\alpha}{1-(1-P)^N}$ and $\frac{\sum_{j=1}^{N} \beta(j) \binom{N}{j} P^j (1-P)^{N-j}}{1-(1-P)^N}$, respectively representing the expected contribution of the fixed cost to the total search cost and the expected cost of obtaining the agent values. The expected number of search rounds is $\frac{1}{1-(1-P)^N}$. The probability of obtaining j agent values is $\frac{\binom{N}{j} P^j (1-P)^{N-j}}{1-(1-P)^N}$, since this becomes a Bernoulli sampling process with a success probability of $1-(1-P)^N$.

The value $P = P^*$ that minimizes $V(P)$ in (4.11) is the optimal reservation probability. If P^* cannot be solved for directly, then it can be solved for by numerical approximation. Based on (4.2), each r_i corresponding to P^* can be calculated by solving for r_i in the equation $P^* = \frac{F(r_i)-F(r_{i-1})}{1-F(r_{i-1})}$, that is, with the equation

$$r_i = F^{-1}(P^*(1-F(r_{i-1})) + F(r_{i-1})) \tag{4.12}$$

We now analyze several special cases of the problem. While the optimal strategies in some of the cases may seem intuitive, their proofs are not necessarily trivial. The first case is when $\beta(i) = 0$, that is, when there is no cost for obtaining the values of the agents. Proposition 4.1 states that the optimal strategy in this case is to use a single search round by setting $r_1 = x_{max}$.

Proposition 4.1. If $\beta(i) = 0$, the optimal strategy is $[x_{max}]$.

Proof. Substituting 0 for $\beta(j)$ in (4.11) obtains $V(P) = \frac{\alpha}{1-(1-P)^N}$. Setting $P = 1$ minimizes this expression, which is equivalent to using x_{max} in the first search round. □

Similarly, Proposition 4.2 states that the optimal strategy when there is no cost for initiating a new search round ($\alpha = 0$) is to increment the reservation value by the smallest amount possible, such that the expected number of values obtained is minimized.

Proposition 4.2. If $\alpha = 0$, the optimal strategy is to use $P \to 0$, i.e., to increment the reservation value by $\varepsilon \to 0$ each round.

Proof. Notice that $lim_{P\to 0}\frac{\beta(j)\binom{N}{j}P^j(1-P)^{N-j}}{1-(1-P)^N}$ equals $\beta(j)$ for $j = 1$ and 0 for $j > 1$, according to L'Hôpital's rule. Substituting $\alpha = 0$ in (4.11), we obtain $lim_{P\to 0}V(P) = \beta(1)$. Since $\beta(i)$ is increasing, according to the model assumption, the result obtained is the minimum expected cost possible. □

Next, we consider the case where the cost of obtaining the values of j agents is linear in j, i.e., $\beta(j) = cj$. This setting is highly applicable as agents, in many

cases, are evaluated individually and independent of one another. Substituting $\beta(j) = cj$, the expression $\sum_{j=1}^{N} j\binom{N}{j}P^j(1-P)^{N-j}$ in the numerator of (4.11) is the mean of a binomially distributed random variable, which equals NP. Therefore,

$$V(P) = \frac{\alpha + cNP}{1-(1-P)^N} \tag{4.13}$$

This result enables the proof of Proposition 4.3, which highlights the nature of the trade-off by which P^* is set.

Proposition 4.3. When $\beta(j)$ is linear in j, the reservation probability that minimizes $V(P)$, $P = P^*$, satisfies $c = (1-P^*)^{N-1}V(P^*)$.

Proof. Differentiating (4.13) with respect to P and setting it to zero obtains:

$$\frac{cN(1-(1-P)^N) - N(1-P)^{N-1}(\alpha + cNP)}{(1-(1-P)^N)^2} = 0 \tag{4.14}$$

Notice that $V(P)(1-(1-P)^N) = \alpha + cNP$ according to (4.13). Substituting the latter expression into (4.14), we observe that the value P^* which satisfies the equation is given by $cN(1-(1-P)^N) - N(1-P)^{N-1}V(P)(1-(1-P)^N) = 0$. Solving for c gets $c = (1-P)^{N-1}V(P)$. □

The explanation of Proposition 4.3 requires understanding the trade-off associated with any increase in P. By increasing P, the chance of finding each of the agents increases. Each agent found due to the increased chance will incur a cost c. The benefit is that if the agent found due to the increase is the only agent found in that round, then the increase actually saves the expected cost $V(P)$ associated with

continuing the search. The probability that the latter case holds is $(1-P)^{N-1}$ (i.e., when all other agents are characterized with a value above the reservation value set using (4.12)). Otherwise, the search just ends. Since the incurred cost c is fixed and the expected benefit $(1-P)^{N-1}V(P)$ decreases as P increases, the optimal P value satisfies $(1-P)^{N-1}V(P) = c$, i.e., when the additional benefit due to the potential saving is offset by the cost incurred by finding that agent.

4.3 Comparative Illustration

In this section, we illustrate the behavior and performance of the optimal strategy derived in the previous section. We show the effect of N, α, and β on the optimal strategy and its associated cost. Additionally, we demonstrate the improvement achieved by the optimal search strategy over several base strategies. The magnitude of improvement in some instances illustrates the importance of choosing the right strategy, while other instances demonstrate how these base strategies can sometimes be close to optimal. Since this problem has not been well addressed in the literature, we adapt three well-studied expanding ring search strategies to our problem [3, 9, 22]. Expanding ring search is used to find routes in ad hoc networking and to locate files in peer-to-peer networking. In this method, the searcher assigns a query a time-to-live (TTL) value, which determines the number of hops the query is forwarded. If the goal is not met, the searcher repeats the query with a larger TTL value. The cost structure of expanding ring search is different than the cost structure of the problem addressed in this paper, since the

cost per round increases with the search extent, and the search extents are typically drawn from a range of discrete values. Still, in the absence of more suitable alternatives, expanding ring-based strategies are a natural basis for comparison to increasing threshold search. In the following paragraphs, we describe the three strategies and then compare their performance in our context.

4.3.1 Two-step strategy

If an increasing threshold search is to improve search costs, then a two-step strategy $S = [r_1, r_2 = x_{max}]$ alone will provide some improvement [9]. The expected total cost when using this strategy is:

$$V(S) = \alpha + \sum_{j=1}^{N} \beta(j) \binom{N}{j} P^j (1-P)^{N-j} + (1-P)^N (\alpha + \beta(N)) \qquad (4.15)$$

in which $P = F(r_1)$. The first two terms are the costs associated with the use of the first reservation value r_1, and the third term is the cost of obtaining all agent values if no values were obtained using r_1. The optimal strategy is obtained by determining the value P that minimizes (4.15).

4.3.2 Fixed increment strategy

A common design of a multi-round expanding ring search strategy is to use a fixed increment between search extents, searching up to some cutoff value before searching the entire search range [22]. The search sequence is thus of the form $\{x_{min} + \delta, x_{min} + 2\delta, \ldots, x_{min} + \mu\delta, x_{max}\}$. The expected cost of using such a se-

quence can be calculated using (4.4). The increment and cutoff value pair with the lowest expected cost is selected from all possible pairs.

4.3.3 California split rule strategy

Another well-studied strategy is the California split rule [3]. According to this strategy, the search extent is doubled each round. We adapt this method to our problem by including a cutoff value above which the reservation value is set to x_{max}, similar to the fixed increment strategy. The search sequence is thus of the form $\{x_{min}+\delta, x_{min}+2\delta, \ldots, x_{min}+2^{\mu}\delta, x_{max}\}$. The expected cost of using such a sequence can be calculated using (4.4). The increment and cutoff value pair with the lowest expected cost is selected from all possible pairs.

4.3.4 Evaluation

Since the domination of the reservation probability based strategy over the different expanding ring based strategies is unquestionable due to its proven optimality, the goal of this evaluation is merely to demonstrate the effect of different parameters on performance. For this purpose, a synthetic environment is an ideal testbed. In the environment used, the agents values are associated with a truncated normal distribution of values [20], with $\mu = 0.5$ and $\sigma = 0.125$, over the interval $(0,1)$; that is, $f(x) = \frac{8\phi(\frac{x-0.5}{0.125})}{\phi(\frac{1-0.5}{0.125})-\phi(\frac{-0.5}{0.125})}$ for $0 \leq x \leq 1$, and 0 otherwise. Note again that the optimal strategy is distribution independent; the probability function only affects the performance of the other strategies. The cost of simul-

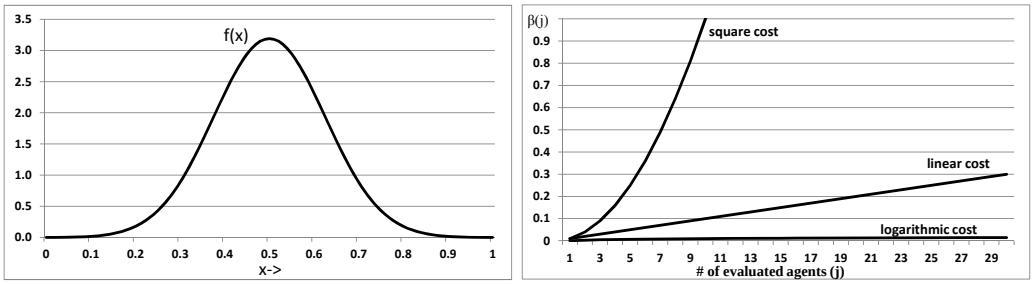


Figure 4.3: Truncated normal distribution function, $f(x)$, with $\mu = 0.5$ and $\sigma = 0.125$ over the interval (0,1) (left) and the three variants of the cost function $\beta(j)$.

Table 4.1: Parameter settings in the synthetic environment

parameter	value
$f(x)$	$\frac{8\phi(\frac{x-0.5}{0.125})}{\phi(\frac{1-0.5}{0.125})-\phi(\frac{-0.5}{0.125})}$
α	$0.001, 0.01, 0.1, 1$
$\beta(j)$	$0.01w(j)$, $w(j) \in \{log(j), j, (j)^2\}$

taneously obtaining the values of j other agents is set to $\beta = 0.01w(j)$, where $w(j) \in \{log(j), j, (j)^2\}$ (denoted "log cost", "linear cost" and "square cost", respectively). The use of the three different cost functions enables testing different rates of increase in the marginal cost of obtaining the value of an additional agent. The cost of initiating a search round and publishing the reservation value is set to $\alpha \in \{0.001, 0.01, 0.1, 1\}$ in order to capture the effects of the magnitude of the ratio $\alpha/\beta(j)$. Table 4.1 summarizes these parameters, and Figure 4.3 illustrates the truncated normal distribution function $f(x)$ and the set of cost functions assigned to $\beta(j)$.

Figure 4.4 shows the expected cost and number of search rounds required to find the best-valued agent as a function of the number of agents N, for the different values of α and function assignments to $\beta(j)$. Note that the scale of the y-axis in

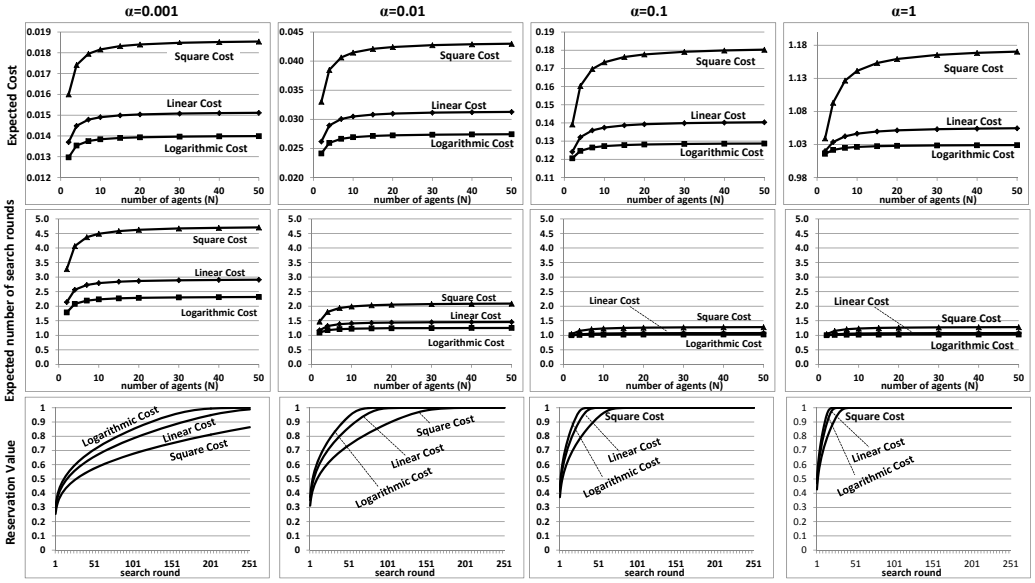


Figure 4.4: Increasing threshold search for the settings in Table 4.1: expected cost (top row); expected number of rounds (middle row); reservation values for $N = 20$ (bottom row)

the first set of charts varies with α in order to improve visualization. The figure also shows the reservation values to be used in each search round when $N = 20$. As expected, α has a substantial effect on the search strategy and expected cost. The reservation value increases as a function of α, since the tradeoff associated with each reservation value accounts for a greater cost of continuing the search if no agent is found. Although the expected number of search rounds decreases as a consequence, the expected cost still increases, due to the greater cost per round. For large values of α and all $\beta(j)$ cost functions, the reservation probability is set such that the expected number of search rounds is close to 1 due to the substantial cost of initiating a new search round.

Amongst the three cost functions, the expected cost is minimal for the log

cost function and maximal for the square cost function, in direct correlation with the order of growth of the three functions. The inverse relationship holds for the reservation values because of the higher cost of finding more than one agent associated with the higher order functions. As a result, the expected cost increases due to the increase in the expected number of rounds.

Another important observation from Figure 4.4 is that the expected cost increases with the number of agents N. Although the expected number of rounds decreases with N for all reservation probabilities, the expected number of agents found also increases. The increase in cost associated with N indicates that, in these settings, the reduction in the number of search rounds does not offset the cost of finding additional agents.

From the graphs of the expected number of search rounds (middle row), we observe that the expected number of search rounds is below five for all combinations of parameters. This supports the applicability of the optimal strategy: Despite the fact that it is an infinite sequence, the search time in practice is comparable to search with competitive finite sequences.

Figure 4.5 shows the expected cost of using the (optimal) reservation probability based strategy and the three expanding ring based strategies as a function of N, for different values of α and function assignments to $\beta(j)$. The graphs illustrate the domination of the optimal strategy and the dominance relationships amongst the remaining strategies. The two step strategy never dominates the fixed increment and California split rule strategies, since it is merely a specific instance of both. Although the fixed increment strategy dominates the California split rule

strategy in these settings, it is possible to construct specific settings in which the latter dominates the first. It is interesting that while the expected cost of the optimal strategy always increases with the number of agents, the expected costs of the fixed increment and California split rule strategies decrease in some settings. This is because the increased probability of finding an agent with each reservation value sometimes reduces the inefficiency imposed by the constraints on the patterns of increase in the reservation value. Another interesting observation is that the expected costs of the fixed increment and California split rule strategies converge as α increases.

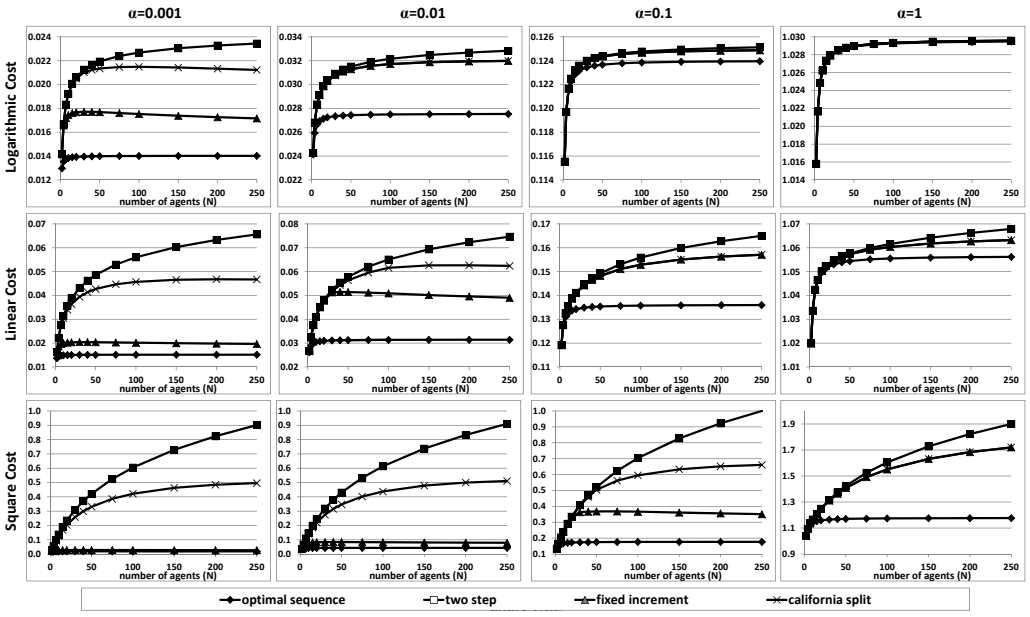


Figure 4.5: Performance comparison of the optimal strategy and the three expanding ring based strategies

The upper row of graphs in Figure 4.5 corresponds to the logarithmic $\beta(j)$ function. All four strategies use higher reservation values as α increases, as the

cost of finding more than one agent in a search round becomes negligible in comparison to the cost of requiring an additional search round. Consequently, the three alternative strategies quickly converge to the same expected cost, as their sequences in those cases contain only two reservation values. The optimally infinite strategy, however, produces observably better results, even for $\alpha = 0.1$. This property, that the California split and fixed increment strategies converge more quickly to the performance of the worst strategy (two step) with respect to α than the optimal solution, occurs for the linear cost (middle row) and the square cost (bottom row) functions as well, for larger values of α. In particular, it is interesting to see that for the square cost function, the fixed increment strategy, which performs quite close to the optimal strategy for small α values, becomes substantially worse as α increases. Overall, in these settings, the improvement of the optimal strategy over the other three methods increases as the order of growth of $\beta(j)$ is increased. Since the optimal search sequence is not constrained to any pattern, it makes a better tradeoff between the cost of finding more than one agent and initiating additional search rounds.

4.4 Multiple Agent Search

In various settings, search for multiple agents is preferable and even necessary. For example, a sink in a sensor network may require readings from several sensors to accurately analyze conditions in the field. In all of these examples, the searcher needs to find several best agents. This section extends the analysis of the single

agent threshold search to a multiple best-valued agents search.

We apply the analysis methodology described in Section 4.2 to the model described in Section 4.1, except that the searcher is interested in finding the K best agents. Without loss of generality, the K best agents are those associated with the K lowest values. The searcher continues its search as long as only $k < K$ agents have been found so far. The searcher's state at the beginning of any round is therefore denoted by the pair (r,k), in which r is the last reservation value the searcher used and k is the number of agents found so far.

A search strategy is defined by a function $S : (r,k) \rightarrow r'$, $r' > r$, in which r' is the reservation value to use in a round with initial state (r,k). Since the reservation value r' depends on the number of agents found so far, a search strategy cannot be defined *a priori* by a sequence of reservation values as in the case of a single agent search. Instead, it can be defined by a decision tree, in which each node represents a state (r,k) and has child nodes for all possible $k \leq k' < K$ total number of agents found by the end of that round. Since many states may be repeated throughout the tree, it can be represented more compactly as a directed graph. The searcher terminates its search when at least $K-k$ agents are found in a round with state (r,k) or when $S(r,k) = x_{max}$, in which case all K agents are necessarily found. It is realistically assumed that agents only reply the first time they comply with a reservation value. Therefore, the searcher only searches among the remaining $N-k$ agents that did not comply with any reservation value less than or equal to r. It is assumed that the publishing cost is constant (α) for all k, as opposed to assuming that the publishing cost is a function of k. The ensuing analysis is

still valid even if these last two assumptions are not made by substituting N for $N-k$ and $\alpha(k)$ for α. Given these assumptions, the expected cost of continuing the search from state (r,k) when using strategy S, denoted $V^{(r,k)}(S)$, is defined recursively as follows:

$$V^{(r,k)}(S) = \alpha + \sum_{j=0}^{N-k} \Big(\beta(j) + V^{(S(r,k),k+j)}(S)\Big) \cdot \binom{N-k}{j} F(S(r,k)|r)^j (1-F(S(r,k)|r))^{N-k-j} \quad (4.16)$$

in which $V^{(r,k)}(S) = 0$ for all $k \geq K$. The total cost of the search is $V^{(x_{min},0)}(S)$.

We again divide the problem into the discrete and continuous cases. In the discrete case, the reservation values can only be selected from a finite set of L values $\{x_1,\ldots,x_L\}$, $x_{min} < x_i < x_{i+1} < x_L = x_{max}$. The optimal strategy can be derived with the following dynamic programming formulation, in which $C(l,k)$ is the cost of continuing the search after using reservation value x_l and after a total of k agents have been found so far. The optimal cost is determined by $C(0,0)$. The

runtime complexity is $O(L^2KN)$.

$$
\begin{aligned}
C(L,k) &= 0 \qquad\qquad \forall 0 \le k \le K \\
C(l,K) &= 0 \qquad\qquad \forall 0 \le l \le L \\
C(l,k) &= \min_{l+1 \le i \le L} \Bigg\{ \alpha + \sum_{j=0}^{N-k} (\beta(j) + C(i,k+j)) \\
&\qquad\qquad \cdot \binom{N-k}{j} F(x_i|x_l)^j (1 - F(x_i|x_l))^{N-k-j} \Bigg\} \\
&\qquad\qquad\qquad\qquad \forall 0 \le k < K, \quad \forall 0 \le l < L \qquad (4.17)
\end{aligned}
$$

For the continuous case, the optimal strategy is not expressed as a single reservation probability as in the single agent search; rather, as the following analysis shows, it is based on a set of reservation probabilities $S = \{P_0, ..., P_{K-1}\}$, where P_k is the reservation probability to be used in any state (r,k), $x_{min} \le r < x_{max}$, $0 \le k < K$. The reservation value r' in state (r,k) can be calculated from the equation $r' = F^{-1}(P_k(1 - F(r)) + F(r))$. As in the single agent search, we prove that this is the optimal strategy by first establishing that the optimal strategy to apply as long as only $k < K$ agents have been found is either an infinite sequence of reservation values or the single reservation value x_{max}.

Theorem 5. The optimal sequence of reservation values to be used from any state (r,k) onwards when $k < K$ and no transition is made to a state (r',k') for any $r' > r$ and $k' > k$ is either $[r_1 = x_{max}]$ or the infinite sequence $[r_1, r_2, \ldots]$, $x_{min} < r_i < x_{max}, \forall i > 0$, where $F(r_i|r_{i-1}) = P_k$, for all $i \ge 1$ and for some P_k.

Proof. We prove this by strong induction on k. Let S^* be the optimal strategy.

Assume that $V^{(r,i)}(S^*) = V_i$ for all $x_{min} \leq r \leq x_{max}$ and some constant V_i, for all $k < i \leq N$. That is, the optimal cost of continuing the search when i items have been found is constant, regardless of the starting reservation value. This is true in the base cases: by definition, $V_i = 0$ for all $i \geq K$. Now, assume that the finite sequence $S_1 = [r_1, \ldots, r_M]$, $r_i > r \; \forall 1 \leq i \leq M$, is the optimal strategy at state (r,k) as long as no transition is made to state (r',k') for any $r' > r$ and $k' > k$. The expected cost of using this sequence is:

$$V^{(r,k)}(S_1) = \sum_{i=1}^{M} \left(\alpha + \sum_{j=1}^{N-k} (\beta(j) + V_{k+j}) \binom{N-k}{j} F(r_i|r_{i-1})^j (1 - F(r_i|r_{i-1}))^{N-k-j} \right) \cdot \prod_{j=1}^{i-1} (1 - F(r_j|r_{j-1}))^{N-k} \quad (4.18)$$

As in Theorem 4, we use $S_2 = [r_2, \ldots, r_M]$ to denote the optimal strategy to be used if no agents are found in the first search round and denote its expected cost from that point on by $V^{(r_1,k)}(S_2)$. Using S_2, we construct strategy $S_1' = [r_2', \ldots, r_M']$ to be applied from the first round, where $F(r_i') = F(r_i|r_1) \; \forall 1 < i \leq M$. The new strategy S_1' has an expected cost $V^{(r,k)}(S_1')$, which equals $V^{(r_1,k)}(S_2)$ according to (4.18). Since S_1 is the optimal strategy, $V^{(r,k)}(S_1) \leq V^{(r,k)}(S_1')$. Now consider a new strategy $S_2' = [r_1'', \ldots, r_M'']$ to be applied from the second round on, where $F(r_i''|r_1) = F(r_i) \; \forall 1 \leq i \leq M$. We denote the expected cost of S_2' from that point on by $V^{(r_1,k)}(S_2')$. According to (4.18), we obtain $V^{(r_1,k)}(S_2') = V^{(r,k)}(S_1)$. Since S_2 is the optimal strategy from the second round, $V^{(r_1,k)}(S_2) \leq V^{(r_1,k)}(S_2')$, result-

ing in $V^{(r,k)}(S_1) \leq V^{(r,k)}(S_1') = V^{(r_1,k)}(S_2) \leq V^{(r_1,k)}(S_2') = V^{(r,k)}(S_1)$, which can hold only if $V^{(r_1,k)}(S_2) = V^{(r,k)}(S_1)$. The same logic can be applied to any search round $j \leq M$, resulting in $V^{(r_{j-1},k)}(S_j) = V^{(r,k)}(S_1)$. In particular, $V^{(r,k)}(S_1) = V^{(r_{M-1},k)}(S_M)$. However, the cost onwards when reaching the last (M^{th}) round, $V^{(r_{M-1},k)}(S_M)$, equals $\alpha + \beta(N-k)$ since $r_M = x_{max}$. Therefore, we obtain $V^{(r,k)}(S_1) = \alpha + \beta(N-k)$, which is equivalent to $S_1 = [r_1 = x_{max}]$. Any optimal strategy that has an expected cost less than $\alpha + \beta(N-k)$ must therefore consist of an infinite sequence and satisfy $F(r_i|r_{i-1}) = F(r_{i+1}|r_i) = P_k$, $\forall i > 0$ and some $0 \leq P_k \leq 1$.

Since the optimal sequence is infinite and the expected cost from each round onwards is stationary, the expected cost of using P_k is:

$$V(P_k) = \alpha + \sum_{j=1}^{N-k} (\beta(j) + V_{k+j}) \binom{N-k}{j} {P_k}^j (1-P_k)^{N-k-j} + V(P_k)(1-P_k)^{N-k} \tag{4.19}$$

Consequently,

$$V(P_k) = \frac{\alpha + \sum_{j=1}^{N-k} (\beta(j) + V_{k+j}) \binom{N-k}{j} {P_k}^j (1-P_k)^{N-k-j}}{1-(1-P_k)^{N-k}} \tag{4.20}$$

The value $P_k = P_k^*$ that minimizes $V(P_k)$ in (4.20) is the optimal reservation probability in any state (r,k). Thus, $V^{(r,k)}(S^*) = V_k$ for some V_k, completing the inductive step. □

The optimal values of P_k ($0 \leq k < K$) can be calculated using backward induction: Given the optimal P_j for all $j > k$, calculate P_k ($0 \leq k < K$) with (4.20). Note

that it is necessary to check in each round if it is better to use $P_k = 1$, as this is also an applicable strategy. The actual reservation values are derived in a similar manner as in the single agent search. The first reservation value r_0 is calculated with $r_0 = F^{-1}(P_0)$. Then, the reservation value to be used in any state (r,k) is calculated by substituting P_k for P^* in (4.12).

We illustrate the properties of the optimal search for multiple agents under the settings in Table 4.1 and compare it to an alternative strategy based on the single agent search. In the alternative strategy, the searcher searches for one agent at a time, using the optimal P from (4.11) for $N-k$ agents when k agents have already been found. That is, the searcher begins by searching for one agent using P from (4.11). Upon finding $k \geq 1$ agents, the searcher begins a new search for one agent using P from (4.11), modified to reflect that only $N-k$ agents are left to reply, and that all remaining agent values are above the last reservation value used. This is continued until the total number of values obtained is greater than or equal to K. Figure 4.6 shows the optimal values of P_k according to (4.20) and P according to (4.11) as functions of the number of agents k already found for $N = 20$; different numbers of agents K that need to be found; $\alpha = 0.1$; and the different $\beta(j)$ cost functions. As can be observed from the figure, P_k decreases as k increases for the log and linear cost functions and all values of K, at a greater rate for the linear cost function. This can be attributed to two conflicting effects of the increase in k on the search: On one hand, as the number of available agents decreases, the expected number of agents found with any reservation value decreases, possibly supporting an increase in the reservation probability. On the other hand, the probability of

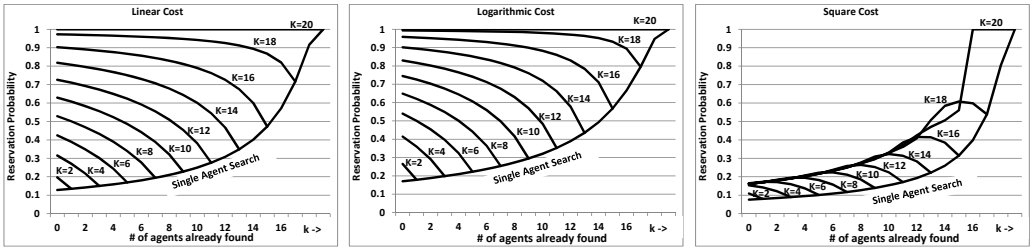


Figure 4.6: The reservation value used by the optimal strategy (P_k) and alternative strategy (P) as a function of k, for different values of K and $N = 20$

finding more than $K - k$ agents with each reservation probability increases, supporting a decrease in the reservation probability. Since the latter effect is more influential than the first, the reservation probability decreases as a function of k.

The behavior of the square cost function differs from the other two. The most obvious difference is that the reservation probability is initially small, increases as the number of agents found increases, then decreases. In particular, when all agents need to be found ($K = 20$), the reservation probability is set to a value less than 1, even though no unnecessary agents will be found if it is set to 1. This is because the cost function is concave; thus, there is a high penalty of finding a large number of agents at once. A careful analysis of the data reveals that the reservation probabilities in this case are indeed set such that the expected number of agents found is almost the same each round. However, as the number of remaining agents approaches 0, the reservation probability decreases accordingly to avoid finding unnecessary agents.

As expected, the optimal reservation probability P_k for the multi agent search is substantially greater than the optimal reservation probability for a single agent

search. In most cases, P_k increases as the total number of agents K that need to be found increases. The square cost function is an exception. As evident from Figure 4.6, the reservation probability for the case ($K = 18$,$k = 14$) is greater than the one for the case ($K = 20$,$k = 14$). This is attributed to the nature of the cost function, as discussed above. In particular, the tradeoff between the expected number of rounds and agents found differs for the case of ($K = 20$,$k = 14$), since in this case there is no problem of finding more agents than necessary.

Figure 4.7 shows the percentage by which the optimal strategy reduces the expected cost of the alternative strategy as a function of K, for different cost functions and values of N. The cost reduction highly depends on the cost function $\beta(j)$ and can be quite substantial, up to 20% for the square cost function and 80% for the log cost function. Furthermore, the improvement increases as the total number of agents K that need to be found increases, as the savings by finding several agents in one round increases. This improvement is mild for the square cost function for the reasons discussed above. It is noteworthy that N has only a minor effect on the cost reduction. An increase in N results in a small decrease in the cost reduction, at a decreasing rate.

4.5 Application to Economic Search

While the goal of increasing threshold search is to find the best-valued agent, the goal of searchers in many applications may be to find a suitable agent while optimizing the process as a whole [4, 23, 45]. For example, consider a buyer

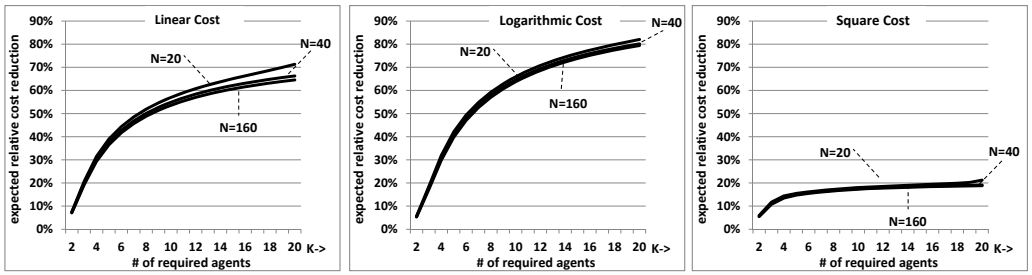


Figure 4.7: Percentage by which the optimal strategy reduces the expected cost of the alternative strategy as a function of K

agent that is interested in purchasing a product and that obtaining posted prices from seller agents incurs a cost (e.g., communication costs). The buyer agent can purchase the product from any of the seller agents. As the buyer increases the number of sellers with which it communicates, the price it expects to pay for the product decreases, but its overall communication costs increase. Thus, the optimal search strategy is a trade-off between the marginal saving of each additional price obtained and the cost of obtaining it.

The research domain in which such problems are studied is called *search theory* ([40, 43], and references therein). Within the framework of search theory, two main clusters of search models can be found: (a) the sequential search model and (b) the fixed sample size model. In the sequential search model [40, 50], the searcher obtains a single agent value at a time, allowing multiple search stages. An example of a sequential search is a buyer who checks prices at different stores until the available options are satisfactory, and then returns to the store with the best option. In the fixed sample size model, the searcher obtains a large set of agent values in a single search round [56], and then chooses the agent associated

with the best value. This is most applicable when some time constraint limits the searcher to a single search round. For example, when applying to college, one must typically apply to several institutions at the same time and then choose from the best offer after all the applications have been reviewed.

While economic search strategies are inapplicable to our problem, as they do not guarantee finding the best-valued agent, increasing threshold search alone and the extensions to it analyzed in this section are useful contributions to economic search theory. This is because increasing threshold search, whenever applicable, can result in an overall reduced cost, even in comparison to the optimal economic search strategy. In this section, we show how the searching agent can benefit from threshold based searches. We begin by introducing the optimal sequential search and fixed sample strategies as known from search theory. We then show how increasing threshold search can be used as an alternative to sequential search, both on its own and in conjunction with sequential search. Finally, we show how the fixed sample search can be augmented with a threshold-based search to improve performance.

4.5.1 Optimal economic search strategies

In the sequential search model [35, 40] a searcher is given N possible opportunities $B = \{B_1, ..., B_N\}$ (e.g., to buy a product) out of which she can choose only one. Each opportunity B_i encapsulates a value to the searcher (e.g., expense, reward, utility). While this value is unknown to the searcher, she is acquainted with the probability distribution function $f(x)$ with which all the values are associated. The

true value of any opportunity B_i can be obtained by paying a fee, denoted c.

The searcher can continue to obtain the value of any of the opportunities in B, paying each time the cost c. Once the searcher decides to terminate her search (or once she has obtained the value of all opportunities), she collects the minimum value among those revealed up until that time (assuming that she seeks the minimum value). The goal of the searcher is therefore to find the optimal strategy, i.e., a stopping rule that minimizes the expected expense, defined as the value eventually obtained plus the accumulated costs incurred throughout the process.

The optimal search strategy for this model is reservation-value based [43]. Note that the term "reservation value" has a slightly different meaning in this context than in the context of increasing threshold search. The searcher sets a reservation value r (i.e., a threshold) and sequentially obtains the value of different opportunities, incurring a cost c, until revealing a value less than the reservation value (or until the values of all opportunities are obtained). This strategy is preferred in particular when the cost of obtaining the value of j agents is linear or super-linear in j, since the searcher does not benefit from obtaining the value of more than one agent at a time. The optimal reservation value, r, is based on the distribution of agent values and the cost of obtaining a value, c [40, 43, 61]. It is the value by which the searcher is indifferent between terminating the search and obtaining r, and resuming the search. The optimal reservation value in this case is derived from [43]:

$$c = \int_{x_{min}}^{x_{max}} (r - \min(y, r) f(y)\, dy = \int_{x_{min}}^{r} F(y)\, dy \tag{4.21}$$

Search theory focuses merely on the optimal stopping rule and does not place much importance on the expected cost of using this rule. Therefore, to compare increasing threshold search with sequential search, we ought to explicitly develop the expected overall cost V of using the latter method. When there are N agents, the expected overall cost of this process is:

$$V = E[X|X \leq r]\sum_{i=1}^{N} F(r)(1-F(r))^{i-1} + E_N[X|X > r](1-F(r))^N + c\sum_{i=1}^{N}(1-F(r))^{i-1}$$
$$= E[X|X \leq r](1-(1-F(r))^N) + E_N[X|X > r](1-F(r))^N + c\frac{1-(1-F(r))^N}{F(r)} \quad (4.22)$$

where $E[X|X \leq r]$ is the expected value of an agent whose value is known to be in the range $[x_{min}, r]$, and $E_N[X|X > r]$ is the expected minimum value in a sample of size N when the minimum value is in the range $[r, x_{max}]$. The first two terms in (4.22) reflect the expected value of the opportunity returned by the search and the third term is the expected cost of obtaining the values. The first term is for when the search terminates with an agent with a value less than r (with probability $\sum_{i=1}^{N} F(r)(1-F(r))^{i-1}$), while the second term is when all of the agent values are above r (with probability $(1-F(r))^N$), in which case the smallest value amongst all N agents is selected. $E[X|X \leq r]$ can be calculated using $f(x|x \leq r) = \frac{f(x)}{F(r)}$ and $F(x|x \leq r) = \frac{F(x)}{F(r)}$, such that $E[X|X \leq r] = \int_{x_{min}}^{r} y\frac{f(y)}{F(r)}\,dy = r - \frac{1}{F(r)}\int_{x_{min}}^{r} F(y)\,dy$. $E_N[X]$ can be calculated using the PDF $f_N(x)$ and CDF $F_N(x)$ of the minimum of

a N-size sample. $F_N(x)$ is the probability that at least one agent in a sample of size N has the value x or less, which can be expressed as $F_N(x) = 1-(1-F(x))^N$. Thus, $E_N[X]$ can be calculated as follows, using integration by parts in the second step:

$$\begin{aligned} E_N[X] &= \int_{x_{min}}^{x_{max}} y f_N(y)\, dy = x_{max} - \int_{x_{min}}^{x_{max}} F_N(y)\, dy \\ &= x_{min} + \int_{x_{min}}^{x_{max}} (1-F(y))^N\, dy \end{aligned} \tag{4.23}$$

$E_N[X|X > r]$ can be calculated by replacing $F(x)$ with $F(x|x > r)$ and x_{min} with r in (4.23).

As opposed to the sequential search, in the fixed sample size model [56], the searcher is limited to the selection of one sample of agents overall in a single period of time. The searcher then selects the agent with the lowest value from this sample. The expected cost of this strategy as a function of the number of agents simultaneously sampled, $0 < K \leq N$, is given by:

$$V = E_K[X] + \beta(K) \tag{4.24}$$

The optimal sample size is the value of K that minimizes (4.24). For general $\beta(i)$, we need to check $K = 1, \ldots N$ for the value with the lowest expected cost. If $\beta(i)$ is linear in i (that is, $\beta(i) = ci$) then it is only necessary to solve for K in the equation $E_K[X] - E_{K+1}[X] = c$ and then choose $\lfloor K \rfloor$ [56].

4.5.2 Increasing threshold-limiting sequential search

Increasing threshold search by itself may be a good alternative to economic search. Since increasing threshold search is only applicable when multiple search rounds are allowed, its most straightforward use is as an alternative to sequential search [40, 50]. The expected cost of the increasing threshold search under the economic search model is composed of the expected cost of search (from (4.11)) and the expected minimum value obtained (from (4.23)):

$$V = \frac{\alpha + \sum_{j=1}^{N} \beta(j) \binom{N}{j} P^j (1-P)^{N-j}}{1-(1-P)^N} + E_N[X] \tag{4.25}$$

The performance of increasing threshold search when used as an alternative to economic search can be improved by processing the agents found using sequential search. According to this improvement, called "increasing threshold-limiting sequential search" from here on, the searcher publishes the thresholds as before until at least one agent responds. Then, instead of processing all of the responses, the searcher processes the responses sequentially, in a random order, as long as the revealed value so far is above some reservation value. In some ways, the searcher is following the original sequential search method; however, its sample space is more targeted, as it is limited by the last threshold to which agents comply. The expected cost of applying a strategy $S = [r_1, \ldots, r_m = x_{max}]$ is thus similar to (4.4), except that $\beta(j)$ is replaced with the cost $V(j, r_{i-1}, r_i)$ of conducting the optimal sequential economic search on j items whose values are between the last two reservation values used, r_{i-1} and r_i. In this case, The-

orem 4 and the resulting solution relying on a fixed reservation probability may no longer be applicable, since the transformation from S_2 to S_1' and like transformations may not be possible. The complexity of deriving the optimal solution is beyond the scope of this paper. Instead, we show how to solve a discrete version of the problem, in which the reservation values are chosen from a finite set $\{x_1,\ldots,x_L\}$, $x_{min} < x_i < x_{i+1} < x_L = x_{max}$, just as in Section 4.2. Replacing $\beta(j)$ with $V(j,x_i,x_l)$ in (4.8) results in the following dynamic programming formulation:

$$C(L) = 0$$

$$C(l) = \min_{l+1\le i\le L} \Bigg\{ \alpha + \sum_{j=1}^{N} V(j,x_l,x_i) \binom{N}{j} F(x_i|x_l)^j (1-F(x_i|x_l))^{N-j} + C(x_i)(1-F(x_i|x_l))^N \Bigg\} \qquad \forall 0 \le l < L \tag{4.26}$$

For any pair of values x_i and x_l, r and $V(j,x_i,x_l)$ can be calculated using (4.21) and (4.22), respectively, replacing $F(x)$ with $\frac{F(x)-F(x_i)}{F(x_l)-F(x_i)}$, x_{min} with x_i, and x_{max} with x_l.

In Figure 4.8, we illustrate the properties and benefits of using increasing threshold search and its improved form as an alternative to sequential search. We use the synthetic environment described in Section 4.3 and Table 4.1, except that agents values are associated with the uniform distribution ($f(x) = 1$ for $0 \le x \le 1$ and $f(x) = 0$ otherwise). Figure 4.8 shows the overall cost of all three strategies (sequential, increasing threshold, and increasing threshold-limiting sequential search) as a function of the number of agents in the environment (horizontal

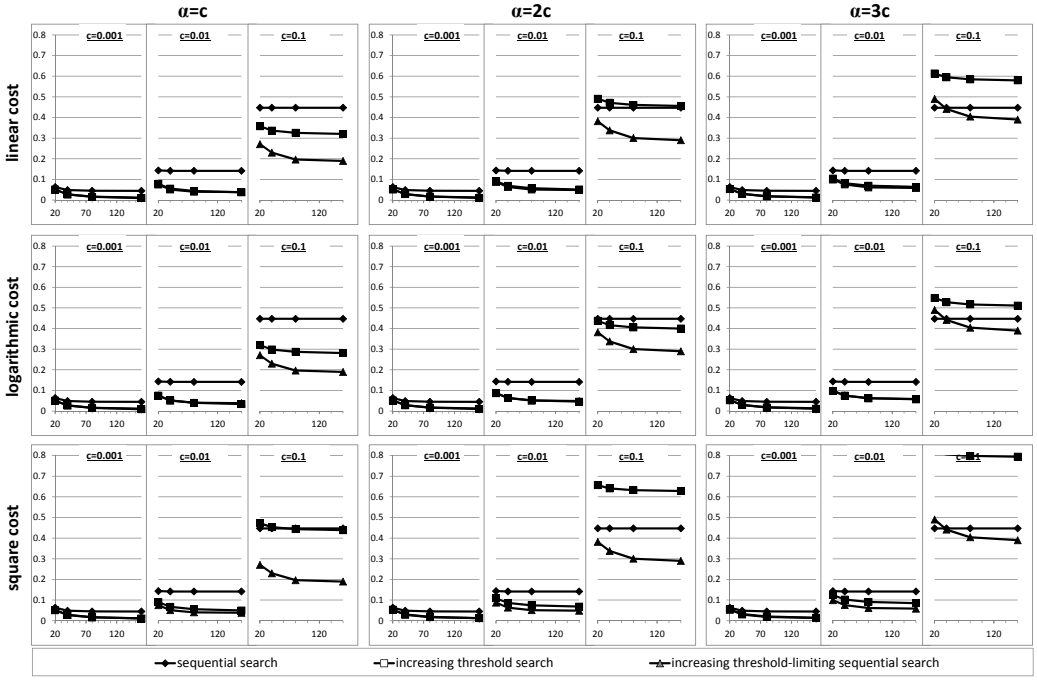


Figure 4.8: Expected overall cost of economic and increasing threshold search

axis), for different values of the ratio $\alpha/\beta(1)$, different cost functions (linear, logarithmic, and square), and different values of the coefficient c in $\beta(i)$. As can be observed from the figure, increasing threshold-limiting sequential search results in a lower overall cost than increasing threshold search, and this improvement increases as a function of α. While this characteristic always holds for linear and concave cost functions, it may not hold when the cost is convex (e.g., logarithmic), since the benefit of evaluating all agents found might outweigh the savings obtained by evaluating only part of the set one by one. Overall, both forms of increasing threshold search perform better than sequential search for low α and c values. In these cases, the improvement in value of the selected agent is greater

than the additional costs α incurred during the search. Whenever the ratio between the fixed cost of each search iteration and the cost of evaluating an agent ($\alpha/\beta(1)$) is sufficiently low, increasing threshold search results in a lower expected overall cost than sequential search. This is a result of the low search costs due to the low $\alpha/\beta(1)$ ratio. As α increases, the advantages of the threshold-based methods diminish because the sequential search does not incur the cost α.

One interesting observation is that while the expected search cost of increasing threshold search increases as a function of the available agents N (see Figure 4.4), its overall cost (search cost and value obtained) decreases as N increases. This is because the reduction in the expected minimum value due to the additional agents outweighs the increased search costs. Likewise, we observe that as N increases, the two increasing threshold-based searches dominate sequential search for large $\alpha/\beta(1)$ ratios.

Finally, we note that the difference between the sequential search and the increasing threshold-limiting sequential search does not depend on the cost function $\beta(j)$. This is because both methods process agent values sequentially, incurring a cost $\beta(1)$ for each agent sample. This property is especially advantageous for these search methods if the cost function $\beta(j)$ is convex (e.g., square cost).

4.5.3 Combined fixed sample size and threshold search

The fixed sample size search [56] can also benefit from threshold based searches. While the restriction to a single search round precludes the use of an increasing threshold search to its full extent, the searcher can still improve expected search

costs by publishing a single reservation value r besides sampling a fixed number of agent values, K. By publishing a reservation value, the searcher reduces the expected minimum value found by sampling alone, although with the additional expected cost of a one round threshold search. Sampling is still necessary, since it is possible that no agents will comply with the published reservation value. Because only one search round is allowed, both the threshold-based sampling and fixed size sampling must be conducted simultaneously. Returning to the example of contract bidding, this is like a case in which the agency must make a decision in the length of time it takes for contractors to prepare and submit their bids. The agency can simultaneously post a call for all bids under some threshold while soliciting bids from select contractors, and then choose the lowest of all bids. The expected cost in this case is:

$$V = \beta(K) + \alpha + \sum_{j=1}^{N-K} \beta(j) \binom{N}{j} F(r)^j (1-F(r))^{N-K-j} + E_N[X|X \leq r] \cdot (1-(1-F(r))^N) + E_K[X|X > r] \cdot (1-F(r))^N \quad (4.27)$$

The first three terms in the above equation reflect the search costs, while the last two terms reflect the expected value of the agent found. The first term is the cost of sampling K elements; the next two terms are for the expected cost of the performing a threshold search on the remaining $N-K$ elements; the fourth term is the expected value of the agent found if its value is below the search threshold r, which will be found either by the sample or the threshold search; and the fifth

term is the expected value of the agent if its value is above r, in which case it is necessarily one of the K agents sampled. $E_N[X|X \leq r]$ can be calculated using $f_N(x|x \leq r) = \frac{f_N(x)}{F_N(r)}$ as follows:

$$E_N[X|X \leq r] = \int_{y=x_{min}}^{r} y \frac{f_N(y)}{F_N(r)} dy = \frac{x_{min} + \int_{y=x_{min}}^{r} (1-F(y))^N dy}{1-(1-F(r))^N} \tag{4.28}$$

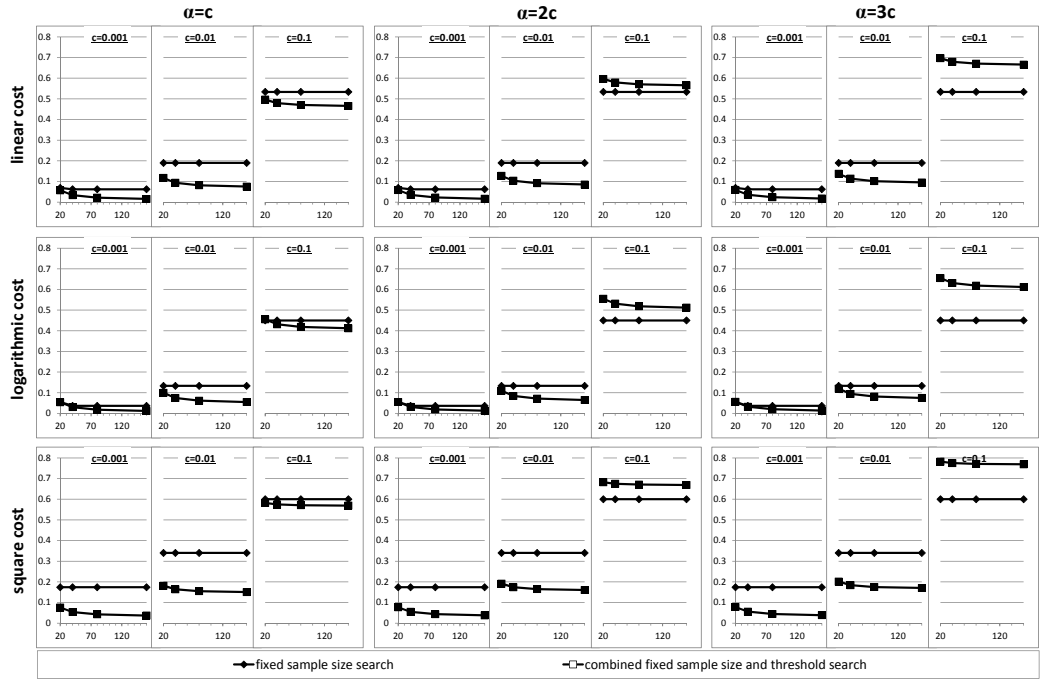


Figure 4.9: Expected overall cost of the fixed sample size and combined fixed sample size and threshold searches as a function of N

Figure 4.9 shows the overall cost of the optimal fixed sample size search and combined fixed sample size and threshold search, using the same settings as in the previous section, as a function of the number of agents in the environment

(horizontal axis) for different values of the ratio $\alpha/\beta(1)$, different cost functions (linear, logarithmic, and square), and different values of the coefficient c in $\beta(i)$. The combined search substantially improves performance when the ratio $\alpha/\beta(1)$ or the value of c is small. In these cases, the improvement from possibly finding a more targeted set of agents (that comply with the threshold) is greater than the additional cost α incurred. As α increases, the benefit of the combined search diminishes, and for large α values, there is no benefit to the combined search. This is also true for large values of c, since the expected number of samples obtained according to the optimal strategy is reduced, resulting in an increase in the expected value obtained. Similar to increasing threshold-limiting sequential search, the accumulated search cost of the combined search decreases as a function of N. In contrast, the difference in cost between the combined search and the fixed sample size search highly depends on the cost function used. The improvement is greater for concave functions, such as the square cost function, because the combined search uses a small fixed sample and sets the reservation value to minimize the number of agents found.

Chapter 5

Conclusion

Effective data collection by wireless sensor networks is challenged by various resource constraints. By choosing the right abstraction, these problems can be addressed as high level optimization problems. For the zone allocation problem, the use of a common result from stochastic geometry, several simplifying assumptions about coverage, and some algebraic manipulation leads to a solution whose runtime is linear with respect to the number of zones. By posing the sensor selection problem as a graph-theoretic problem and relating it to the budgeted and generalized maximum coverage problems, it is possible to derive a solution with a tight approximation bound and prove that it is the best possible approximation factor amongst all possible polynomial time solutions. Framing the best-valued data problem as a multi-agent search problem leads to the derivation of an optimal solution that is distribution independent.

The dual of the coverage problem is to minimize the number of sensors re-

quired for an expected level of coverage. Chapter 2 shows how to calculate this directly using a closed-form expression. The dual of the selection problem is to minimize the budget required to meet quality of information goal. Although this problem is not addressed in Chapter 3, it is closely related to minimum set cover. Unlike the maximum coverage problems, the best possible approximation factor for minimum set cover is $\log n$. Indeed, the derivation of the $(e-1)/e$ bound on the budgeted maximum coverage is based on this result. Consequently, minimizing the budget for sensor selection also has poor approximation bounds.

Without loss of generality, a sub-classification of minimization problems are minimax and min-sum problems, in which the objective is to minimize the maximum value and to minimize the total value, respectively. Minimax problems address fairness constraints, such as guaranteeing limits on the largest areas left uncovered and the largest difference between predicted and actual values. Only the "max-sum" versions of the coverage and selection problems were addressed. The greedy selection algorithm, in its most general form, can be used to address minimax problems by changing the objective function. Since a minimax objective function is submodular, the same approximation bounds apply according to Sviridenko [58].

The analysis of the zone allocation problem is based on a commonly used formula for expected coverage. In practice, actual coverage may be different than expected because of border effects. While it is possible to derive more accurate formulas, the simulation studies showed that the common formula can be effectively used to derive the optimal allocation when zones are sufficiently large. The

expected coverage when zones are too small to ignore border effects still needs analysis. In reality, the zones themselves may have diverse conditions, but if there is a good approximation of the expected coverage within a zone, a similar method of deriving the optimal allocation can be applied. Another issue to consider is coverage by distributions other than the uniform distribution. The simulations modeled environmental effects on sensing ranges with weighted distance functions. This opens up some interesting new areas of studies. An interesting problem is to determine optimal sensor placements under this model. Since this is most likely a hard problem, an alternative goal is find bounds on achievable coverage by different placement strategies.

Prior work on deterministic deployment modeled coverage with grids, as described in Chapter 1. By modeling coverage with grids, the coverage problem is effectively the same as the selection problem. As such, the greedy algorithm cannot guarantee coverage greater than $(e-1)/e)$ of the optimal. More problematic is the runtime, as demonstrated in Chapter 2. Although the main analysis of zone allocation was for random deployment, it was originally designed to overcome the inefficiencies of greedy placement. By realistically assuming that the field can be divided into zones of similar conditions, the sensors can first be allocated to each zone and then deployed within each zone according to well-studied strategies for homogeneous fields. The shortcoming of this approach is that it does not adequately account for border effects and that optimal arrangements in bounded regions are difficult to derive. If the field cannot be divided into large enough zones such that border effects cannot be ignored, then this technique is no longer

effective. However, a divide-and-conquer approach can be designed instead. By bounding the difference between coverage of a zone by some arrangement of sensors and optimal coverage of that zone, the field can be divided in a way that minimizes the error.

Several natural generalizations are possible for the problem of sensor selection. (1) The information network can be more generally cast as a hypergraph rather than a simple graph. In such a case, there can be an edge between any set of vertices rather than a single pair of vertices. The problem again grows exponentially as the allowable number of vertices in a set increases, so some limitation on the sets considered must be placed. (2) In the problem considered, the link types are homogeneous. In more general cases, different links may be of different types, and it may be possible to make type specific predictions. For example, the amount of computation can be reduced by performing the regression analysis on a small subset of links of each type. (3) The predicted sensor data is only as good as the task it is used for. In particular, the data may only be used to detect various events. Selecting sensors according to their ability to predict other streams may not be the best set for overall event detection. The right model for prediction and selection must therefore be considered in this case.

Increasing threshold search is applicable to many multi-agent systems. For example, a volunteer ambulance corps dispatcher needs to find the closest volunteer to an emergency. She must page the volunteers and request that they call back to learn their locations. Instead of requesting that all volunteers call back, she can request that only volunteers within a certain distance of the emergency call back,

and repeat the request with greater distances until at least one volunteer calls. Another example is a government agency seeking bids for a project. It may request a best and final offer from a set of the top bidders after an initial call for bids, so it would try to limit its initial search to only those top bidders.

The analysis of increasing threshold search is for the case when the distribution of agents' values is known to the searcher. Just as for expanding ring search, the optimal increasing threshold search strategy in the discrete case can be derived using dynamic programming. The optimal strategy in the continuous case can be derived using the unique probabilistic properties described in Chapter 4. What remains is a study of the case when the distribution of agents' values is unknown to the searcher. Just as for expanding ring search, it may be possible to prove the competitive ratio of different strategies and bound the best possible competitive ratio of any strategy.

Appendix A

Proof of Proposition 2.8

Proof. This is evident when $i = 1$ by setting $C_1 = 1$ and $\alpha'_1 = \alpha_1$. For $i \geq 2$ zones, begin with the assumption that the optimal coverage of Z'_{i-1} by n sensors is equivalent to Eq. (2.15), such that the expected coverage under a partition β'_i is formulated as

$$\sum_{j=1}^{i-1} \gamma_j \left(1 - C_{i-1} e^{-\frac{\alpha'_{i-1} S}{\sum_{j=1}^{i-1} \gamma_j A} (1-\beta'_i) n} \right) + \gamma_i \left(1 - e^{-\frac{\alpha_i S}{\gamma_i A} \beta'_i n} \right) \tag{A.1}$$

Taking the second derivative of Eq. (A.1) with respect to β'_i shows that it is concave. Therefore, there is one maximum for any value of β'_i. A sketch of the derivation of the optimal value of β'_i is as follows. Set the first derivative of Eq. (A.1) with respect to β'_i equal to 0 and solve for β'_i.

$$\beta'_i = \frac{\alpha'_{i-1} \gamma_i}{\alpha'_{i-1} \gamma_i + \alpha_i \sum_{j=1}^{i-1} \gamma_j} \left(\frac{\sum_{j=1}^{i-1} \gamma_j A}{\alpha'_{i-1} S n} \ln \frac{\alpha_i}{\alpha'_{i-1} C_{i-1}} + 1 \right) \tag{A.2}$$

Substitute this expression into Eq. (A.1) and reduce to derive an expression for coverage equivalent to Eq. (2.15)

$$\sum_{j=1}^{i} \gamma_j \left(1 - \frac{\sum_{j=1}^{i-1} \gamma_j \alpha_i + \alpha'_{i-1} \gamma_i}{\sum_{j=1}^{i} \gamma_j \alpha'_{i-1}} \cdot \left(\frac{\alpha'_{i-1} C_{i-1}}{\alpha_i} \right)^{\frac{\alpha_i \sum_{j=1}^{i-1} \gamma_j}{\alpha'_{i-1} \gamma_i + \alpha_i \sum_{j=1}^{i-1} \gamma_j}} e^{-\frac{\alpha_i \alpha'_{i-1} \sum_{j=1}^{i} \gamma_j}{\alpha'_{i-1} \gamma_i + \alpha_i \sum_{j=1}^{i-1} \gamma_j} \frac{Sn}{\sum_{j=1}^{i} \gamma_j A}} \right) \quad \text{(A.3)}$$

in which α'_i and C_i are defined recursively as follows:

$$\begin{aligned}
\alpha'_1 &= 1 \\
\alpha'_{i \geq 2} &= \frac{\alpha_i \alpha'_{i-1} \sum_{j=1}^{i} \gamma_j}{\alpha'_{i-1} \gamma_i + \alpha_i \sum_{j=1}^{i-1} \gamma_j} \\
C_1 &= 1 \\
C_{i \geq 2} &= \frac{\sum_{j=1}^{i-1} \gamma_j \alpha_i + \alpha'_{i-1} \gamma_i}{\sum_{j=1}^{i} \gamma_j \alpha'_{i-1}} \left(\frac{\alpha'_{i-1} C_{i-1}}{\alpha_i} \right)^{\frac{\alpha_i \sum_{j=1}^{i-1} \gamma_j}{\alpha'_{i-1} \gamma_i + \alpha_i \sum_{j=1}^{i-1} \gamma_j}}
\end{aligned}$$

The complete derivation of Eq. (A.2) can be found in Appendix B.4. □

Appendix B

Derivations

B.1 β_{opt}, Eq. (2.6) in Section 2.2

Recall that the expected coverage of the field is

$$1 - \gamma_1 e^{-\frac{\alpha_1 S}{\gamma_1 A}(1-\beta)n} - \gamma_2 e^{-\frac{\alpha_2 S}{\gamma_2 A}\beta n}$$

Take the derivative with respect to β and reduce

$$-\frac{\alpha_1 S}{A} n e^{-\frac{\alpha_1 S}{\gamma_1 A}(1-\beta)n} + \frac{\alpha_2 S}{A} n e^{-\frac{\alpha_2 S}{\gamma_2 A}\beta n}$$

Set the derivative equal to zero, rearrange terms, and eliminate like terms:

$$\alpha_2 e^{-\frac{\alpha_2 S}{\gamma_2 A}\beta n} = \alpha_1 e^{-\frac{\alpha_1 S}{\gamma_1 A}(1-\beta)n}$$

Take the log of both sides of the equation:

$$\ln \alpha_2 - \frac{\alpha_2 S}{\gamma_2 A}\beta n = \ln \alpha_1 - \frac{\alpha_1 S}{\gamma_1 A}(1-\beta)n$$

Solve for β:

$$\frac{\alpha_1 S}{\gamma_1 A}\beta n + \frac{\alpha_2 S}{\gamma_2 A}\beta n = \ln\frac{\alpha_2}{\alpha_1} + \frac{\alpha_1 S}{\gamma_1 A}n$$

$$\frac{Sn}{A}\left(\frac{\alpha_1}{\gamma_1} + \frac{\alpha_2}{\gamma_2}\right)\beta = \ln\frac{\alpha_2}{\alpha_1} + \frac{\alpha_1 S}{\gamma_1 A}n$$

$$\frac{\gamma_1\gamma_2}{\alpha_1\alpha_2}\frac{Sn}{A}\left(\frac{\alpha_1}{\gamma_1} + \frac{\alpha_2}{\gamma_2}\right)\beta = \frac{\gamma_1\gamma_2}{\alpha_1\alpha_2}\left(\ln\frac{\alpha_2}{\alpha_1} + \frac{\alpha_1 S}{\gamma_1 A}n\right)$$

$$\frac{Sn}{A}\left(\frac{\gamma_1}{\alpha_1} + \frac{\gamma_2}{\alpha_2}\right)\beta = \frac{\gamma_2}{\alpha_2}\left(\frac{\gamma_1}{\alpha_1}\ln\frac{\alpha_2}{\alpha_1} + \frac{S}{A}n\right)$$

$$\beta = \frac{1}{\frac{\gamma_1}{\alpha_1} + \frac{\gamma_2}{\alpha_2}}\frac{\gamma_2}{\alpha_2}\left(\frac{A}{Sn}\frac{\gamma_1}{\alpha_1}\ln\frac{\alpha_2}{\alpha_1} + 1\right)$$

Note that $\alpha_1 = 1$ by definition, so $\ln\frac{\alpha_2}{\alpha_1} = \ln\alpha_2$.

B.2 Two zone coverage, Eq. (2.8) in Section 2.2

When $n \le -\frac{A}{S}\frac{\gamma_1}{\alpha_1}\ln\alpha_2$, all sensors are assigned to Z_1, so the expected coverage is:

$$\gamma_1\left(1 - e^{-\frac{\alpha_1 S}{\gamma_1 A}n}\right)$$

To derive an expression for the coverage when $n > -\frac{A}{S}\frac{\gamma_1}{\alpha_1}\ln\alpha_2$, substitute β_{opt} into Eq. (2.5) and simplify

$$\begin{aligned}
&1 - \gamma_1 e^{-\frac{\alpha_1 S}{\gamma_1 A}\left(1 - \frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{\gamma_2}{\alpha_2}\left(\frac{A}{Sn}\frac{\gamma_1}{\alpha_1}\ln\alpha_2 + 1\right)\right)n} - \gamma_2 e^{-\frac{\alpha_2 S}{\gamma_2 A}\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{\gamma_2}{\alpha_2}\left(\frac{A}{Sn}\frac{\gamma_1}{\alpha_1}\ln\alpha_2 + 1\right)n} \\
&= 1 - \gamma_1 e^{-\frac{\alpha_1}{\gamma_1}\frac{Sn}{A}} e^{\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{\gamma_2}{\alpha_2}\ln\alpha_2} e^{\frac{\alpha_1}{\gamma_1}\frac{Sn}{A}\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{\gamma_2}{\alpha_2}} - \gamma_2 e^{-\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{\gamma_1}{\alpha_1}\ln\alpha_2} e^{-\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{Sn}{A}} \\
&= 1 - \left(\gamma_1 e^{-\frac{\alpha_1}{\gamma_1}\frac{Sn}{A}} e^{\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{\gamma_2}{\alpha_2}\ln\alpha_2} e^{\frac{\alpha_1}{\gamma_1}\frac{Sn}{A}\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{\gamma_2}{\alpha_2}} e^{\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{\gamma_1}{\alpha_1}\ln\alpha_2} e^{\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{Sn}{A}} + \gamma_2\right) \\
&\quad \cdot e^{-\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{\gamma_1}{\alpha_1}\ln\alpha_2} e^{-\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{Sn}{A}}
\end{aligned}$$

$$
\begin{aligned}
&= 1 - \left(\gamma_1 e^{\left(-\frac{\alpha_1}{\gamma_1}\left(\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}\right)+1+\frac{\alpha_1}{\gamma_1}\frac{\gamma_2}{\alpha_2}\right)\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{Sn}{A}} e^{\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\left(\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}\right)\ln\alpha_2} + \gamma_2 \right) \\
&\quad \cdot e^{-\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{\gamma_1}{\alpha_1}\ln\alpha_2} e^{-\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{Sn}{A}} \\
&= 1 - (\gamma_1\alpha_2+\gamma_2)\,\alpha_2^{-\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{\gamma_1}{\alpha_1}} e^{-\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{Sn}{A}} \\
&= 1 - \left(\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}\right)\alpha_2\alpha_2^{-\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{\gamma_1}{\alpha_1}} e^{-\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{Sn}{A}} \\
&= 1 - \left(\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}\right)\left(\alpha_2^{\frac{\gamma_2}{\alpha_2}}\right)^{\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}} e^{\frac{1}{\frac{\gamma_1}{\alpha_1}+\frac{\gamma_2}{\alpha_2}}\frac{Sn}{A}}
\end{aligned}
$$

B.3 Two zone minimum sensor count, Eq. (2.7) in Section 2.2

Just as the formula for C is conditional on whether $n \leq -\frac{A}{S}\frac{\gamma_1}{\alpha_1}\ln\alpha_2$ or not, the formula for the minimum number of sensors n to achieve an expected level of coverage C depends on whether or not C is less than or equal to the expected coverage when $n = -\frac{A}{S}\frac{\gamma_1}{\alpha_1}\ln\alpha_2$. The expected coverage for this value of n is:

$$
\begin{aligned}
&\gamma_1\left(1 - e^{\frac{\alpha_1 S}{\gamma_1 A}\frac{A}{S}\frac{\gamma_1}{\alpha_1}\ln\alpha_2}\right) \\
&= \gamma_1(1-\alpha_2)
\end{aligned}
$$

To derive a formula for n when $C = \gamma_1\left(1 - e^{-\frac{\alpha_1 S}{\gamma_1 A}n}\right)$, rearrange the terms:

$$e^{-\frac{\alpha_1 S}{\gamma_1 A}n} = \frac{\gamma_1 - C}{\gamma_1}$$

Take the log of both sides:

$$-\frac{\alpha_1 S}{\gamma_1 A}n = \ln\frac{\gamma_1 - C}{\gamma_1}$$

Solve for n:

$$n = -\frac{\gamma_1 A}{\alpha_1 S} \ln \frac{\gamma_1 - C}{\gamma_1}$$

To derive a formula for n when $C = 1 - \left(\frac{\gamma_1}{\alpha_1} + \frac{\gamma_2}{\alpha_2}\right) \left(\alpha_2^{\frac{\gamma_2}{\alpha_2}}\right)^{\frac{1}{\frac{\gamma_1}{\alpha_1} + \frac{\gamma_2}{\alpha_2}}} e^{\frac{1}{\frac{\gamma_1}{\alpha_1} + \frac{\gamma_2}{\alpha_2}} \frac{Sn}{A}}$, rearrange the terms:

$$1 - C = \left(\frac{\gamma_1}{\alpha_1} + \frac{\gamma_2}{\alpha_2}\right) \left(\alpha_2^{\frac{\gamma_2}{\alpha_2}}\right)^{\frac{1}{\frac{\gamma_1}{\alpha_1} + \frac{\gamma_2}{\alpha_2}}} e^{\frac{1}{\frac{\gamma_1}{\alpha_1} + \frac{\gamma_2}{\alpha_2}} \frac{Sn}{A}}$$

Take the log of both sides:

$$\ln(1 - C) = \ln \left(\frac{\gamma_1}{\alpha_1} + \frac{\gamma_2}{\alpha_2}\right) + \frac{1}{\frac{\gamma_1}{\alpha_1} + \frac{\gamma_2}{\alpha_2}} \frac{\gamma_2}{\alpha_2} \ln \alpha_2 + \frac{1}{\frac{\gamma_1}{\alpha_1} + \frac{\gamma_2}{\alpha_2}} \frac{Sn}{A}$$

Solve for n:

$$n = \frac{A}{S} \left(\frac{\gamma_1}{\alpha_1} + \frac{\gamma_2}{\alpha_2}\right) \left(\ln(1 - C) - \ln \left(\frac{\gamma_1}{\alpha_1} + \frac{\gamma_2}{\alpha_2}\right) - \frac{1}{\frac{\gamma_1}{\alpha_1} + \frac{\gamma_2}{\alpha_2}} \frac{\gamma_2}{\alpha_2} \ln \alpha_2\right)$$

B.4 β'_{opt_i}, Eq. (A.2) in Section 2.3

Recall that the expected coverage of Z'_i under a partition β'_i of n sensors is reformulated as:

$$\sum_{j=1}^{i-1} \gamma_j \left(1 - C_{i-1} e^{-\frac{\alpha'_{i-1} S}{\Sigma_{j=1}^{i-1} \gamma_j A} (1 - \beta'_i) n}\right) + \gamma_i \left(1 - e^{-\frac{\alpha_i S}{\gamma_i A} \beta'_i n}\right)$$

Take the derivative with respect to β'_{opt_i} and reduce

$$-\frac{\alpha'_{i-1} Sn}{A} C_{i-1} e^{-\frac{\alpha'_{i-1} S}{\Sigma_{j=1}^{i-1} \gamma_j A} (1 - \beta'_i) n} + \frac{\alpha_i Sn}{A} e^{-\frac{\alpha_i S}{\gamma_i A} \beta'_i n}$$

Set it equal to 0, rearrange the terms, and eliminate like terms

$$\alpha_i e^{-\frac{\alpha_i S}{\gamma_i A} \beta'_i n} = \alpha'_{i-1} C_{i-1} e^{-\frac{\alpha'_{i-1} S}{\Sigma_{j=1}^{i-1} \gamma_j A} (1 - \beta'_i) n}$$

Take the log of both sides

$$\ln \alpha_i - \frac{\alpha_i S}{\gamma_i A}\beta_i' n = \ln(\alpha_{i-1}' C_{i-1}) - \frac{\alpha_{i-1}' S}{\sum_{j=1}^{i-1}\gamma_j A}(1-\beta_i')n$$

Rearrange the terms and solve for β_i'

$$\frac{\alpha_{i-1}' S}{\sum_{j=1}^{i-1}\gamma_j A}\beta_i' n + \frac{\alpha_i S}{\gamma_i A}\beta_i' n = \ln \alpha_i - \ln(\alpha_{i-1}' C_{i-1}) + \frac{\alpha_{i-1}' S}{\sum_{j=1}^{i-1}\gamma_j A} n$$

$$\frac{\alpha_{i-1}'\gamma_i + \alpha_i \sum_{j=1}^{i-1}\gamma_j}{\gamma_i \sum_{j=1}^{i-1}\gamma_j}\frac{Sn}{A}\beta_i' = \frac{\alpha_{i-1}' Sn}{\sum_{j=1}^{i-1}\gamma_j A}\left(\frac{\sum_{j=1}^{i-1}\gamma_j A}{\alpha_{i-1}' Sn}\ln\frac{\alpha_i}{\alpha_{i-1}' C_{i-1}} + 1\right)$$

$$\beta_i' = \frac{\alpha_{i-1}'\gamma_i}{\alpha_{i-1}'\gamma_i + \alpha_i \sum_{j=1}^{i-1}\gamma_j}\left(\frac{\sum_{j=1}^{i-1}\gamma_j A}{\alpha_{i-1}' Sn}\ln\frac{\alpha_i}{\alpha_{i-1}' C_{i-1}} + 1\right)$$

To solve for C_i and α_i', first expand $(1-\beta_i')$:

$$1-\beta_i' = 1 - \frac{\alpha_{i-1}'\gamma_i}{\alpha_{i-1}'\gamma_i + \alpha_i \sum_{j=1}^{i-1}\gamma_j}\left(\frac{\sum_{j=1}^{i-1}\gamma_j A}{\alpha_{i-1}' Sn}\ln\frac{\alpha_i}{\alpha_{i-1}' C_{i-1}} + 1\right)$$

$$= \frac{\alpha_{i-1}'\gamma_i + \alpha_i \sum_{j=1}^{i-1}\gamma_j}{\alpha_{i-1}'\gamma_i + \alpha_i \sum_{j=1}^{i-1}\gamma_j} - \frac{\alpha_{i-1}'\gamma_i}{\alpha_{i-1}'\gamma_i + \alpha_i \sum_{j=1}^{i-1}\gamma_j}\frac{\sum_{j=1}^{i-1}\gamma_j A}{\alpha_{i-1}' Sn}\ln\frac{\alpha_i}{\alpha_{i-1}' C_{i-1}} - \frac{\alpha_{i-1}'\gamma_i}{\alpha_{i-1}'\gamma_i + \alpha_i \sum_{j=1}^{i-1}\gamma_j}$$

$$= \frac{\alpha_i \sum_{j=1}^{i-1}\gamma_j}{\alpha_{i-1}'\gamma_i + \alpha_i \sum_{j=1}^{i-1}\gamma_j} - \frac{\alpha_{i-1}'\gamma_i}{\alpha_{i-1}'\gamma_i + \alpha_i \sum_{j=1}^{i-1}\gamma_j}\frac{\sum_{j=1}^{i-1}\gamma_j A}{\alpha_{i-1}' Sn}\ln\frac{\alpha_i}{\alpha_{i-1}' C_{i-1}}$$

Now substitute $1-\beta_i'$ and β_i' into (A.1) and reduce.

$$\sum_{j=1}^{i}\gamma_j - \sum_{j=1}^{i-1}\gamma_j C_{i-1} e^{-\frac{\alpha_{i-1}' Sn}{\sum_{j=1}^{i-1}\gamma_j A}\left(\frac{\alpha_i \sum_{j=1}^{i-1}\gamma_j}{\alpha_{i-1}'\gamma_i + \alpha_i \sum_{j=1}^{i-1}\gamma_j} - \frac{\alpha_{i-1}'\gamma_i}{\alpha_{i-1}'\gamma_i + \alpha_i \sum_{j=1}^{i-1}\gamma_j}\frac{\sum_{j=1}^{i-1}\gamma_j A}{\alpha_{i-1}' Sn}\ln\frac{\alpha_i}{\alpha_{i-1}' C_{i-1}}\right)}$$

$$- \gamma_i e^{-\frac{\alpha_i S}{\gamma_i A}\frac{\alpha_{i-1}'\gamma_i}{\alpha_{i-1}'\gamma_i + \alpha_i \sum_{j=1}^{i-1}\gamma_j}\left(\frac{\sum_{j=1}^{i-1}\gamma_j A}{\alpha_{i-1}' Sn}\ln\frac{\alpha_i}{\alpha_{i-1}' C_{i-1}} + 1\right)n}$$

$$= \sum_{j=1}^{i} \gamma_j - \sum_{j=1}^{i-1} \gamma_j C_{i-1} e^{-\frac{\alpha_i \alpha'_{i-1} Sn}{A(\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j)}} e^{\frac{\alpha'_{i-1}\gamma_i}{\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j} \ln \frac{\alpha_i}{\alpha'_{i-1} C_{i-1}}}$$

$$- \gamma_i e^{-\frac{\alpha_i \Sigma_{j=1}^{i-1}\gamma_j}{\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j} \ln \frac{\alpha_i}{\alpha'_{i-1} C_{i-1}}} e^{-\frac{Sn}{A} \frac{\alpha_i \alpha'_{i-1}}{\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j}}$$

$$= \sum_{j=1}^{i} \gamma_j - \left(\sum_{j=1}^{i-1} \gamma_j C_{i-1} e^{-\frac{\alpha_i \alpha'_{i-1} Sn}{A(\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j)}} e^{\frac{\alpha'_{i-1}\gamma_i}{\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j} \ln \frac{\alpha_i}{\alpha'_{i-1} C_{i-1}}} \right.$$

$$\left. \cdot e^{\frac{\alpha_i \Sigma_{j=1}^{i-1}\gamma_j}{\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j} \ln \frac{\alpha_i}{\alpha'_{i-1} C_{i-1}}} e^{\frac{Sn}{A} \frac{\alpha_i \alpha'_{i-1}}{\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j}} + \gamma_i \right)$$

$$e^{-\frac{\alpha_i \Sigma_{j=1}^{i-1}\gamma_j}{\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j} \ln \frac{\alpha_i}{\alpha'_{i-1} C_{i-1}}} e^{-\frac{Sn}{A} \frac{\alpha_i \alpha'_{i-1}}{\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j}}$$

$$= \sum_{j=1}^{i} \gamma_j - \left(\sum_{j=1}^{i-1} \gamma_j C_{i-1} e^{\frac{\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j}{\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j} \ln \frac{\alpha_i}{\alpha'_{i-1} C_{i-1}}} + \gamma_i \right)$$

$$\cdot \left(\frac{\alpha'_{i-1} C_{i-1}}{\alpha_i} \right)^{\frac{\alpha_i \Sigma_{j=1}^{i-1}\gamma_j}{\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j}} e^{-\frac{\alpha_i \alpha'_{i-1} \Sigma_{j=1}^{i}\gamma_j}{\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j} \frac{Sn}{\Sigma_{j=1}^{i}\gamma_j A}}$$

$$= \sum_{j=1}^{i} \gamma_j - \left(\sum_{j=1}^{i-1} \gamma_j C_{i-1} \frac{\alpha_i}{\alpha'_{i-1} C_{i-1}} + \gamma_i \right)$$

$$\cdot \left(\frac{\alpha'_{i-1} C_{i-1}}{\alpha_i} \right)^{\frac{\alpha_i \Sigma_{j=1}^{i-1}\gamma_j}{\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j}} e^{-\frac{\alpha_i \alpha'_{i-1} \Sigma_{j=1}^{i}\gamma_j}{\alpha'_{i-1}\gamma_i + \alpha_i \Sigma_{j=1}^{i-1}\gamma_j} \frac{Sn}{\Sigma_{j=1}^{i}\gamma_j A}}$$

$$= \sum_{j=1}^{i} \gamma_j - \frac{\sum_{j=1}^{i-1} \gamma_j \alpha_i + \alpha'_{i-1}\gamma_i}{\alpha'_{i-1}} \left(\frac{\alpha'_{i-1} C_{i-1}}{\alpha_i} \right)^{\frac{\alpha_i \sum_{j=1}^{i-1} \gamma_j}{\alpha'_{i-1}\gamma_i + \alpha_i \sum_{j=1}^{i-1} \gamma_j}} e^{-\frac{\alpha_i \alpha'_{i-1} \sum_{j=1}^{i} \gamma_j}{\alpha'_{i-1}\gamma_i + \alpha_i \sum_{j=1}^{i-1} \gamma_j} \frac{Sn}{\sum_{j=1}^{i} \gamma_j A}}$$

$$= \sum_{j=1}^{i} \gamma_j \left(1 - \frac{\sum_{j=1}^{i-1} \gamma_j \alpha_i + \alpha'_{i-1}\gamma_i}{\sum_{j=1}^{i} \gamma_j \alpha'_{i-1}} \cdot \left(\frac{\alpha'_{i-1} C_{i-1}}{\alpha_i} \right)^{\frac{\alpha_i \sum_{j=1}^{i-1} \gamma_j}{\alpha'_{i-1}\gamma_i + \alpha_i \sum_{j=1}^{i-1} \gamma_j}} e^{-\frac{\alpha_i \alpha'_{i-1} \sum_{j=1}^{i} \gamma_j}{\alpha'_{i-1}\gamma_i + \alpha_i \sum_{j=1}^{i-1} \gamma_j} \frac{Sn}{\sum_{j=1}^{i} \gamma_j A}} \right)$$

Appendix C

Variables Used in Chapter 4

Table C.1: Table of Variables Used in Increasing Threshold Search Analysis

symbol	explanation
N	number of agents in the system
$f(x)$	PDF (probability density function) of agent values
$F(x)$	CDF (cumulative distribution function) of agent values
$[x_{min}, x_{max}]$	range of agent values
$\beta(j)$	cost of obtaining j agent values
r	published reservation (threshold) value
$S = [r_1, r_2, \ldots]$	search strategy
α	fixed cost of publishing a reservation value
$V(S)$	expected cost of using strategy S
S^*	optimal strategy
$f(x\|r_{i-1})$	PDF of agent values if all values are above r_{i-1}
$F(x\|r_{i-1})$	CDF of agent values if all values are above r_{i-1}
$\{x_1, \ldots, x_L\}$	discretized set of potential reservation values
$C(i)$	expected cost of continuing a search after publishing reservation value x_i
c	cost of obtaining the value of one agent in the linear cost model
P	reservation probability used in the optimal strategy
	continued on next page

Table of Variables – continued	
symbol	explanation
δ	increment of reservation value in the fixed increment and California split rule strategies
μ	parameter defining the number of search rounds in the fixed increment and California split rule strategies
K	number of agents that the searcher is interested in finding in a multi agent search; sample size in the fixed sample size economic search model
k	number of agents found so far
(r,k)	state at the beginning of a multi-agent search round
$V^{(r,k)}(S)$	expected cost of continuing a search from state (r,k) when using strategy S
P_i	reservation probability to use when i agents have already been found
$C(l,k)$	expected cost of continuing a multi-agent search after using reservation value x_l and a total of k agents have been found so far
$\{B_1,...,B_N\}$	possible opportunities in the sequential search model
$E[X\|X \leq r]$	expected value of an agent whose value is known to be in the range $[x_{min}, r]$
$E_N[X\|X > r]$	expected minimum value in a sample of size N when the minimum value is in the range $[r, x_{max}]$
$f_N(x)$	PDF of the minimum of a N-size sample
$F_N(x)$	CDF of the minimum of a N-size sample
$f_N(x\|x \leq r)$	PDF of the minimum of a N-size sample when the minimum is less than r

Bibliography

[1] C. Aggarwal. *Data Streams: Models and Algorithms*. Springer, 2007.

[2] Y. Bakos. Reducing buyer search costs: Implications for electronic marketplaces. *Mgmt. Science*, 42:1676–92, 1997.

[3] Y. M. Baryshnikov, E. G. Coffman Jr., P. R. Jelenkovic, P. Momcilovic, and D. Rubenstein. Flood search under the california split rule. *Oper. Res. Lett.*, 32(3):199–206, 2004.

[4] J. Benhabib and C. Bull. Job search: The choice of intensity. *Journal of Political Economy*, 91:747–764, 1983.

[5] P. Brass. Bounds on coverage and target detection capabilities for models of networks of mobile sensors. *TOSN*, 3(2), 2007.

[6] P. Brass, W. O. J. Moser, and J. Pach. *Research Problems in Discrete Geometry*. Springer, 2005.

[7] N. B. Chang and M. Liu. Revisiting the TTL-based controlled flooding search: optimality and randomization. In Z. J. Haas, S. R. Das, and R. Jain, editors, *MOBICOM*, pages 85–99. ACM, 2004. ISBN 1-58113-868-7.

[8] N. B. Chang and M. Liu. Controlled flooding search in a large network. *IEEE/ACM Trans. Netw.*, 15(2):436–449, 2007.

[9] Z. Cheng and W. B. Heinzelman. Flooding strategy for target discovery in wireless networks. *Wireless Networks*, 11(5):607–618, 2005.

[10] M. Chhabra, S. Das, and D. Sarne. Expert-mediated search. In *AAMAS*, pages 415–422, 2011.

[11] S. P. M. Choi and J. Liu. Markov decision approach for time-constrained trading in electronic marketplace. *International Journal of Information Technology and Decision Making*, 1(3):511–524, 2002.

[12] R. L. Church and C. ReVelle. The maximal covering location problem. *Papers in Regional Science*, 32(1):101–118, 1974.

[13] R. Cohen and L. Katzir. The generalized maximum coverage problem. *Inf. Process. Lett.*, 108(1):15–22, 2008.

[14] O. Daescu and J. D. Palmer. Minimum separation in weighted subdivisions. *Int. J. Comput. Geometry Appl.*, 19(1):33–57, 2009.

[15] S. S. Dhillon and K. Chakrabarty. Sensor placement for effective coverage and surveillance in distributed sensor networks. In *WCNC*, volume 3, pages 1609–1614, March 2003.

[16] U. Feige. A threshold of ln for approximating set cover. *J. ACM*, 45(4): 634–652, 1998.

[17] S. Gal, M. Landsberger, and B. Levykson. A compound strategy for search in the labor market. *International Economic Review*, 22(3):597–608, 1981.

[18] M. L. Ginsberg and W. D. Harvey. Iterative broadening. *Artif. Intell.*, 55(2): 367–383, 1992.

[19] D. Golovin, M. Faulkner, and A. Krause. Online distributed sensor selection. In *IPSN*, pages 220–231, 2010.

[20] W. H. Greene. *Econometric Analysis*. Prentice Hall, 5 edition, 2003.

[21] P. Hall. *Introduction to the Theory of Coverage Processes*. John Wiley and Sons, 1988.

[22] J. Hassan and S. Jha. On the optimization trade-offs of expanding ring search. In N. Das, A. Sen, S. K. Das, and B. P. Sinha, editors, *IWDC*, volume 3326 of *Lecture Notes in Computer Science*, pages 489–494. Springer, 2004. ISBN 3-540-24076-4.

[23] N. Hazon, Y. Aumann, and S. Kraus. Collaborative multi agent physical search with probabilistic knowledge. In C. Boutilier, editor, *IJCAI*, pages 167–174, 2009.

[24] D. S. Hochbaum and A. Pathria. Analysis of the greedy approach in problems of maximum k-coverage. *Naval Research Logistics*, 45(6):615–627, 1998.

[25] W. Hu, N. Bulusu, C. T. Chou, S. Jha, A. Taylor, and V. N. Tran. Design and evaluation of a hybrid sensor network for cane toad monitoring. *TOSN*, 5(1), 2009.

[26] J. Kephart and A. Greenwald. Shopbot economics. *JAAMAS*, 5(3):255–287, 2002.

[27] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70(1):39–45, 1999.

[28] S. Kim, S. Pakzad, D. E. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In T. F. Abdelzaher, L. J. Guibas, and M. Welsh, editors, *IPSN*, pages 254–263. ACM, 2007.

[29] R. E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artif. Intell.*, 27(1):97–109, 1985.

[30] H. Koskinen. On the coverage of a random sensor network in a bounded domain. In *Proceedings of 16th ITC Specialist Seminar*, pages 11–18, 2004.

[31] A. Krause and C. Guestrin. Near-optimal observation selection using submodular functions. In *AAAI*, pages 1650–1654, 2007.

[32] A. Krause, C. Guestrin, A. Gupta, and J. M. Kleinberg. Near-optimal sensor placements: maximizing information while minimizing communication cost. In J. A. Stankovic, P. B. Gibbons, S. B. Wicker, and J. A. Paradiso, editors, *IPSN*, pages 2–10. ACM, 2006. ISBN 1-59593-334-4.

[33] A. Krause, A. P. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[34] G. L. Lan, Z. M. Ma, and S. S. Sun. Coverage problem of wireless sensor networks. In J. Akiyama, W. Y. C. Chen, M. Kano, X. Li, and Q. Yu, editors, *CJCDGCGT*, volume 4381 of *Lecture Notes in Computer Science*, pages 88–100. Springer, 2005. ISBN 978-3-540-70665-6.

[35] M. Landsberger and D. Peled. Duration of offers, price structure, and the gain from search. *Journal of Economic Theory*, 16(1):17–37, 1977.

[36] L. Lazos and R. Poovendran. Stochastic coverage in heterogeneous sensor networks. *TOSN*, 2(3):325–358, 2006.

[37] R. Lentz. Sorting by search intensity. *Journal of Economic Theory*, 145(4): 1436 – 1452, 2010. ISSN 0022-0531.

[38] J. Leskovec. Tutorial summary: Large social and information networks: opportunities for ml. In *ICML*, page 179, 2009.

[39] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. VanBriesen, and N. S. Glance. Cost-effective outbreak detection in networks. In *KDD*, pages 420–429, 2007.

[40] S. Lippman and J. McCall. The economics of job search: A survey. *Economic Inquiry*, 14:155–189, 1976.

[41] B. Liu and D. Towsley. A study of the coverage of large-scale sensor networks. In *MASS*, pages 475–483, October 2004.

[42] J. McHugh and Y. Perl. Best location of service centers in a treelike network under budget constraints. *Discrete Mathematics*, 86(1-3):199–214, 1990.

[43] J. McMillan and M. Rothschild. Search. In R. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, pages 905–927. 1994.

[44] N. Megiddo, E. Zemel, and S. L. Hakimi. The maximum coverage location problem. *SIAM Journal on Algebraic and Discrete Methods*, 4(2):253–261, 1983.

[45] P. Morgan and R. Manning. Optimal search. *Econometrica*, 53(4):923–944, 1985.

[46] G. Moscarini and R. Wright. Introduction to search theory and applications. *Journal of Economic Theory*, 145(4):1319 – 1324, 2010. ISSN 0022-0531.

[47] A. Ostfeld(et.al.). The battle of the water sensor networks (bwsn): A design challenge for engineers and algorithms. *Journal of Water Resources Planning and Management*, 134(6):556–568, 2008.

[48] D. Pompili, T. Melodia, and I. F. Akyildiz. Three-dimensional and two-dimensional deployment analysis for underwater acoustic sensor networks. *Ad Hoc Networks*, 7(4):778–790, 2009.

[49] C. ReVelle and R. Swain. Central facilities location. *Geographical Analysis*, 2:30–42, 1970.

[50] M. Rothschild. Searching for the lowest price when the distribution of prices is unknown. *Journal of Political Economy*, 82(4):689–711, 1974.

[51] Y. Sakurai, S. Papadimitriou, and C. Faloutsos. Braid: Stream mining through group lag correlations. In F. Özcan, editor, *SIGMOD Conference*, pages 599–610. ACM, 2005. ISBN 1-59593-060-4.

[52] S. Shakkottai, R. Srikant, and N. B. Shroff. Unreliable sensor grids: coverage, connectivity and diameter. *Ad Hoc Networks*, 3(6):702–716, 2005.

[53] S. Shukla, N. Bulusu, and S. Jha. Cane-toad monitoring in kakadu national park using wireless sensor networks. In *Proceedings of APAN*, 2004.

[54] L. Smith. Frictional matching models. *Annual Reviews in Economics*, 3: 319–338, 2011.

[55] M. E. Stickel and M. Tyson. An analysis of consecutively bounded depth-first search with applications in automated deduction. In *IJCAI*, pages 1073–1075, 1985.

[56] G. Stigler. The economics of information. *Journal of Political Economy*, 69(3):213–225, 1961.

[57] J. Sun, S. Papadimitriou, and C. Faloutsos. Online latent variable detection in sensor networks. In *ICDE*, pages 1126–1127. IEEE Computer Society, 2005. ISBN 0-7695-2285-8.

[58] M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.*, 32(1):41–43, 2004.

[59] D. C. Verma, C. W. Wu, T. Brown, A. Bar-Noy, S. Shamoun, and M. Nixon. Location dependent heuristics for sensor coverage planning. volume 6981, page 69810B. SPIE, 2008.

[60] D. C. Verma, C. W. Wu, T. Brown, A. Bar-Noy, S. Shamoun, and M. Nixon. Application of halftoning algorithms to location dependent sensor placement. In *ISCAS*, pages 161–164, 2009.

[61] M. L. Weitzman. Optimal search for the best alternative. *Econometrica*, 47 (3):641–54, May 1979.

[62] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and yield in a volcano monitoring sensor network. In *OSDI*, pages 381–396. USENIX Association, 2006.

[63] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2):18–25, 2006.

[64] M. Wybo, J. Robert, and P.-M. Léger. Using search theory to determine an applications selection strategy. *Information & Management*, 46(5):285–293, 2009.

[65] G. Xing, R. Tan, B. Liu, J. Wang, X. Jia, and C.-W. Yi. Data fusion improves the coverage of wireless sensor networks. In K. G. Shin, Y. Zhang, R. Bagrodia, and R. Govindan, editors, *MOBICOM*, pages 157–168. ACM, 2009. ISBN 978-1-60558-702-8.

[66] Y. Yang, I.-H. Hou, J. C. Hou, M. Shankar, and N. S. V. Rao. Sensor placement for detecting propagative sources in populated environments. In *INFOCOM*, pages 1206–1214. IEEE, 2009.

[67] B.-K. Yi, N. Sidiropoulos, T. Johnson, H. V. Jagadish, C. Faloutsos, and A. Biliris. Online data mining for co-evolving time sequences. In *ICDE*, pages 13–22, 2000.

[68] Y. Zou and K. Chakrabarty. Uncertainty-aware and coverage-oriented deployment for sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):788–798, 2004.