Dynamic Sensor Tasking in Heterogeneous, Mobile Sensor Networks

by

Peter Jones

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2005

Author ....[illegible]........
Department of Electrical Engineering and Computer Science
May 6, 2005

Certified by ....[illegible]........
Sanjoy Mitter
Professor
Thesis Supervisor

Accepted by ....[illegible]........
Arthur C. Smith
Chairman, Department Committee on Graduate Students

[redacted]

Dynamic Sensor Tasking in Heterogeneous, Mobile Sensor Networks

by

Peter Jones

Submitted to the Department of Electrical Engineering and Computer Science
on May 6, 2005, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Modern sensor environments often attempt to combine several sensors into a single sensor network. The nodes of this network are generally heterogenous and may vary with respect to sensor complexity, sensor operational modes, power costs and other salient features. Optimization in this environment requires considering all possible sensor modalities and combinations. Additionally, in many cases there may be a time critical objective, requiring sensor plans to be developed and refined in real-time. This research will examine and expand on previous work in multi-sensor dynamic scheduling, focusing on the issue of near optimal sensor-scheduling for real-time detection in highly heterogeneous networks.

First, the issue of minimum time inference is formulated as a constrained optimization problem. The principles of dynamic programming are applied to the problem. A network model is adopted in which a single "leader" node makes a sensor measurement. After the measurement is made, the leader node chooses a successor (or chooses to retain network leadership). This model leads to an index rule for leader/action selection under which the leader is the sensor node with maximum expected rate of information acquisition. In effect, the sensor and modality with the maximum ratio of expected entropic decrease to measurement time is shown to be an optimal choice for leader.

The model is then generalized to include networks with simultaneously active sensors. In this case the corresponding optimization problem becomes prohibitively difficult to solve, and so a game theoretic approach is adopted in order to balance the preferences of the several sensors in the network. A novel algorithm for multi-player coordination is developed that uses iterative partial utility revelation to achieve bounded Pareto inefficiency of the solution.

Thesis Supervisor: Sanjoy Mitter
Title: Professor

Acknowledgments

The author would like to acknowledge the following individuals without whose contributions this thesis would not exist:

Professor Sanjoy Mitter for advice, suggestions, readings and encouragement

Dr. Ali Yegulalp for his service as Staff mentor and liason with Lincoln Laboratory

The Lincoln Scholars Committee and Director's Office for supporting the research financially as well as through regular feedback and suggestions

Dr. Ed Baranoski for suggesting the problem of cooperative sensing in the first place

Jason Williams for suggestions, discussions and frequent help with details

Above all my wife who helped when she didn't want to, encouraged when I was frustrated, listened when she'd heard it all before and cared when I needed it the most.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Sensor Networks

A sensor network is any combination of two or more sensors capable of interacting either directly through sensor-to-sensor communication or indirectly through centralized information sharing. Sensor networks come in many configurations and levels of complexity. They may consist of a handful of individual sensors or several thousand (or more). They may be homogeneous (populated by identical sensors) or heterogenous (populated by two or more types of sensors). The sensors themselves can range from the simple, such as passive, single-mode magnetometers to the extremely complex, such as airborne synthetic aperture radars (SARs) capable of operating in a variety of modes and the ability to direct sensing to specific areas through beam forming and radar steering.

But regardless of size, configuration, heterogeneity and sensor complexity, all sensor networks face similar issues. All sensor networks are implemented with the goal of observing the environment, taking measurements and infering some world state from this information. Raw data collected by the sensor nodes must somehow be culled and collated to form a unified estimate of the world state. A problem of converting raw data into information is commonly called an **inference problem**. Some examples of inference problems in sensor networks include the detection, tracking and identification of targets of interest.

1.1.1 General Sensor Network Issues

No network has infinite resources. All networks, whether simple or complex, must realize a resource management strategy in order to accomplish the sensing goals such as target detection, tracking and indentification. Besides limitations due to the number of sensors available within the network, other constraints may include time, energy, material cost, or communication. These constraints will be collectively termed **costs**. Achieving sensing goals while minimizing costs requires careful planning and resource management.

Resource management can be significantly complicated by network heterogeneity. If the sensor network is homogeneous, meaning all sensors have essentially the same capabilities, sensor management becomes arbitrary to a certain extent. In homogeneous networks, the network ability to achieve its goal is invariant under the particular sensor chosen for each sensing task. In heterogeneous networks, however, each sensor has unique abilities and limitations. This diversity disrupts the the network assignment invariance. The number of possible combinations for scheduling can quickly overwhelm the planning capabilities.

1.1.2 Examples of Sensor Networks

Sensor networks are becoming increasingly popular in military and industrial applications. Listed below are several examples of modern sensor networks, all dramatically different in structure, but essentially similar in the purpose of trying to accurately determine the current state of an environment of interest.

Example 1: Walmart recently announced that it will encourage all suppliers to attatch RFID tags to their products. RFIDs are very simple sensors with no power source. They use the energy of a satellite radio wave to accomplish their task, which is to geolocate themselves. This is an example of a homogeneous sensor network, where all sensors are essentially identical. The resources involved include the time it takes a tag to respond to queries about its position as well as the energy cost of transmitting the position. In this network, the inference problem is to determine where an item is at any given time. Using the RFID sensor network to solve this problem will result in greater efficiency in Walmart's supply chain, and lower overall product costs for its consumers.

Example 2: The Advanced Network Technologies division of the National Institue of Standards and Technology has ongoing research regarding the use of sensor networks for search and rescue operations. The networks would function in environments unsuitable to humans, enabling search and rescue workers to penetrate hazardous environments and locate rescue candidates. Such a system was deployed by a team from the University of South Florida following the collapse of the World Trade Center buildings. A wirelessly connected group of moblie robots, each with several sensors attatched, were used in the search the collapsed buildings for trapped victims. In this case, the networks was small scale (5-10 robots) and mostly homogeneous (all robots were similarly constructed and had similar sensing abilities). The goal of identifying rescue candidates is time critical, since the longer the delay the less successful eventual rescue will be. Therefore the primary cost to the network was time.

Example 3: The scenario of interest for this thesis, which will be described in more depth in Section 1.3.1, is that of using sensor networks for battlefield awareness. In this case, the set of sensors is heterogeneous and range from simple acoustic sensors to highly complicated, satellite-based radars. Sensors in the network differ in their level of mobility, their available modes of operation, and their ability to detect targets of interest. The inference problems in this network include target localization, tracking, and identification. The costs in this network are varied, including communication, fuel expenditure, sensor detection and interference by the enemy and more. However, as with the search and rescue example, in many cases the primary concern is one of time, in that targets must be isolated and identified as soon as possible.

The above examples represent a small sampling of interesting sensor networks. For a more complete discussion of areas in which sensor networks are being utilized, see [13]. While the inference problems solved by sensor networks can vary as greatly as the networks themselves, the fundamental issues of resource management and achieving goals under cost constraints are relevant to all real sensor networks.

1.2 Scheduling in Sensor Networks

Sensor networks generally must develop a sensor plan or schedule in order to achieve sensing goals such as target detection, tracking and identification. The sensor plan determines which resources should be used, in what order and for how long. The sensor plan will directly determine how much time, energy and material cost the network must use in order to achieve the sensing goals. The purpose of the scheduler is to create a sensor plan which will minimize the cost required to solve the inference problem. This can be represented by the equation

$$p^* = \arg\min_{p \in P} C(p) \tag{1.1}$$

where $C(p)$ is the cost (in resources) of executing plan p and P is the set of all plans that result in the solution of the inference problem. A **plan** p consists of an ordered set of **actions**, $a_1 ... a_N$. In the case of sensor networks, the actions are queries or deployments of specific sensors. Each action may involve querying or deploying any subset of the sensor network.

Despite the simplicity of the optimization equation, the job of the scheduler is complicated by several issues, including the dynamics of the network, the stochastic nature of sensor measurements, and the uncertainty in calculations of cost, current knowledge state or both.

1.2.1 Static and Dynamic Scheduling

Algorithms for scheduling fall into two broad classes: static (open-loop) and dynamic (closed-loop). Static scheduling is much simpler, but is limited in applicability because it cannot react to unexpected changes to the system. Dynamic scheduling, while significantly more complicated, is generally more robust due to its ability to adapt during the course of plan execution.

In open-loop scheduling, the scheduler determines a complete plan prior to the beginning of the plan execution. The plan cannot be modified once execution has begun. Open-loop scheduling has the benefit of simplicity and ease, and in deterministic systems can guarantee achievement of the system goal. However, static programming suffers from an inability to react to information obtained during the execution stage. Additionally, in stochastic systems open-loop scheduling generally cannot guarantee the eventual achievement of the system goal, since the results of actions are inherently unpredictable.

Closed-loop, or dynamic, scheduling involves interleaving decisions with actions, enabling the system to make scheduling decisions based on all information obtained up to the current time. In this case, the scheduler can react to information obtained during plan execution in order to ensure that appropriate actions are taken at each new step of execution.

Dynamic scheduling depends on the principle of optimality, first identified by Bellman [3]. This principle states that if a path from a start state to a goal state is optimal, then the subpath from any intermediate state to the goal state will be optimal for the subproblem of moving from the intermediate state to the goal. Using this principle we can modify 1.1 to read

$$\begin{aligned} a_i^* &= \arg\min_{a\in A_i}\left(C(a) + C(p_{i+1}^*)\right) \\ a_N^* &= \arg\min_{a\in A_N} C(a) \end{aligned} \tag{1.2}$$

and $p_i^* = \{a_j^*\}_{j=i}^N$. $C(a)$ is commonly refered to as the immediate cost and $C(p_{i+1}^*)$ as the cost-to-go. Programs formulated as 1.2 are called dynamic programs and the method of solving them is called dynamic programming.

1.2.2 Scheduling in Deterministic Systems

It can be shown (e.g. [5]) that in deterministic systems the optimal open-loop schedule is equivalent to the optimal closed-loop schedule. In a degenerate sensor network where all costs are known *a priori* and sensor measurements have no random element, there is no need for a scheduler to be reactive, since no unexpected information will ever occur.

Consider for instance a network populated by sensors, $s_{\{1:M\}}$. Suppose the inference problem for the network is to achieve a certain level L of confidence under some measure. Each sensor contributes a deterministic "amount" of confidence l_i at a cost c_i and the total confidence is $\sum_{i=1}^{N} l_{a_i}$ and the total cost is $\sum_{i=1}^{N} c_{a_i}$. This problem can be mapped to a generalized knapsack problem which is exactly solvable prior to the beginning of execution (although not in polynomial time). We will return to this degenerate formulation in Section 2.4.1.

1.2.3 Scheduling in Stochastic Systems

In the case of stochastic systems, the optimal open-loop schedule will not generally be equivalent to the optimal closed-loop schedule. In stochastic systems, total cost can be improved by reacting to information gained during plan execution. From the above example, if the "amount" of confidence from sensor i is non-deterministic, the shortest path problem becomes a stochastic shortest path problem. Any static plan in this system would suffer from an inability to react to an unexpected l_i. Say, for instance that in creating the static plan, the planner anticipated that after the first k measurements the total level of confidence $\sum_{i=1}^{k} l_i$ would be equal to L and the task would be complete. The optimal plan would not schedule any more actions, since the goal would have been achieved and more actions would only increase the total cost (assuming $c_i > 0 \ \forall i$). If, however, the actual level after k measurements is $\tilde{L} < L$, a scenario which may occur due to the stochastic nature of l_i, the optimal static plan cannot schedule more measurements, and the sensing goal will therefore not be achieved.

Scheduling in stochastic systems can be extemely complicated and computationally intensive. The formalization of DP (dynamic programming) requires the consideration of all possible action sequences that result in achievement of the goal state. This can be improved somewhat by advanced techniques, but the inherent complexity is invariant to such techniques. This is particularly problematic for programs where the set of possible actions is large. For the sensor network, the number of possible actions at each decision stage is combinatorial in the number of sensors, resulting in a very large action set. In complicated stochastic networks methods of approximating the optimal solution are of great importance. Some approximate methods are suggested in Section 1.3.4.

1.3 Proposed Methodology

1.3.1 Scenario

One interesting application of sensor networks is in the area of battlefield awareness. Battlefield awareness is the ability of a military unit to accurately model the current field of engagement, especially as regards the detection, tracking and identification of targets of interest. This awareness is central to the success of military campaigns.

The proliferation of possible sensors for use in battlefield awareness has dramatically increased the ability of military units to accurately model the field of engagement. The challenge is to determine from among all the possible sensor deployments, which deployment will achieve awareness goals while minimizing total system costs. To better define this problem we must define the "awareness goals" of the system as well as the system costs.

1.3.2 Measures of Information

One of the central problems of information theory is how to measure the information content of a model. Examples of this problem include measuring how accurately a quantized number represets a continuous number or how well an estimated probability function represents a true probability function. The latter example is relevant to the question of measuring the information in a sensor network tasked with target detection. In this case, if we let X be the set of possible target locations, the true probability function is

$$p(x) = \begin{cases} 1 & \text{if } x \in X \\ 0 & \text{otherwise} \end{cases} \tag{1.3}$$

Knowledge of this true state is obtained through a series of observations which are aggregated into a conditional probability function $p(x|\mathbf{z})$, where $\mathbf{z}$ is the set of observations. Information theoretic methods, including Shannon entropy and relative entropy, can be applied to $p(x|\mathbf{z})$ to measure how well $p(x|\mathbf{z})$ approximates the true probability function in Equation 1.3.

Let $\mathbf{z} = \{z_i\}_{i=1}^{N}$ be the set of collected sensor measurements. Then, under the assumption that sensor measurements are independent of one another, we can formulate a Bayesian estimate of $p(x)$ as

$$p(x|\mathbf{z}) = \alpha * p_0(x) \prod_{i=1}^{N} p_i(z_i|x) \tag{1.4}$$

where $p_0(x)$ is the *a priori* probability that $x \in X$, N is the number of sensor measurements, $p_i(z_i|x)$ is the conditional probability of sensor measurement z_i given $x \in X$ and α is a scaling constant.

There are several possible measures of the information contained in $p(x|\mathbf{z})$. Perhaps the most well known is the Shannon entropy, due to Claude Shannon, the father of information theory. Shannon postulated that the information in $p(x|z)$ is related to the uncertainty in $p(x|z)$. He then showed that the uncertainty of $p(x|z)$ can be measured using the negative of the Boltzmann entropy, or

$$\mu(p(x|\mathbf{z})) = -\int p(x|\mathbf{z}) log(p(x|\mathbf{z}))dx$$

Thus, one way of formulating the inference problem is to declare it solved once the entropy is driven below a threshold, i.e. $\mu(p(x|\mathbf{z})) \leq \gamma$.

1.3.3 Minimum Time Formulation

In the case of battlefield awareness, there are several possible cost formulations for a particular sensor plan. In this thesis we will focus on the time costs of the system.

Thus, the optimization equation 1.1 can be written

$$p^* = \arg\min_{p\in P} E[t(p)] \tag{1.5}$$

The expectation $E[t(p)]$ is necessary when the exact length of time between the deployment of a sensor and the availability of its measurement is uncertain. For most sensors, at the time of deployment only an expected time or range of times until measurements are available is known.

Additional uncertainty comes from the stochastic nature of the sensors themselves. Without loss of generality we can restrict our attention to sensors for which it is not possible to know beforehand what a particular sensor measurement will be. (Informally, if a sensor's measurement were predetermined and it was included in a sensor plan, the sensor plan could be improved by removing the planned measurement, since the sensor measurement cannot decrease system uncertainty and can only increase the total time of plan execution).

Since the system is stochastic due to uncertainties both in the cost function and in the future knowledge states (i.e. sensor measurements), a dynamic scheduler will be able to achiever higher levels of optimality than a static scheduler (see Section 1.2). We can reformulate the time minimization problem in the form of Equation 1.2

$$a_i^* = \arg\min_{a\in A_i} \left(E[t(a) + t(p_{i+1}^*)]\right) \tag{1.6}$$

and $p_i^* = \{a_j^*\}_{j=i}^N$ and N is the total number of actions in the sensor plan. This states that the optimal action at any given time is the one that minimizes the time it takes to accomplish the action plus the expected time it will take to achieve the system goal once the action has been taken.

1.3.4 Approximate Dynamic Programming

In solving for the optimal action a_i^*, two quantities must be calculated: the immediate cost and the expected cost-to-go. Solving for the optimal cost-to-go involves constructing an optimal plan from the time the current action completes until the system goal is met. This subproblem can become extremely complicated. Even in simple scenarios it can be computationally prohibitive to solve this problem exactly. Frequently, approximate methods must be introduced in order to decrease the complexity of the problem.

The simplest approximate method is to assume the cost-to-go is negligble. This leads to algorithms which minimize immediate costs with no thought for future benefit. These algorithms are generally called **greedy** algorithms. In the case of the formulation we are considering, the greedy algorithm would result in a plan according to the equation

$$\tilde{a}_i = \arg\min_{a \in A_i} E[t(a)]$$

However, if we assume the null action (i.e. no sensors deployed) is always an option, this greedy formulation will result in a plan in which no action is ever taken! Although other greedy formulations can mitigate this effect, they all suffer from the problem of neglecting the contribution of future states to the total cost of a sensor plan.

Another possibility is to approximate $E[t(p_i^*)]$ by some easily computable heuristic function $H(p_i)$. Heuristic methods can significantly reduce the complexity of the sensor planning problem itself, but introduce the problem of choosing which heuristic to use. In general, the closer a heuristic comes to the ideal cost-to-go the better the resulting algorithm performs. Simulation based methods and neuro-dynamic programming are two heuristic methods that have proven effective at finding approximatly optimal solutions while limiting computational complexity.

1.4 Simulated Experiments

Experiments were performed to verify the theoretical developments of this thesis. As stated in Section 1.3.1 the scenario of primary interest is one of detecting battlefield targets. The experiments were designed to imitate features of the battlefield awareness problem and consist of both simulated sensors and simulated environments.

1.4.1 Simulated Environment

The experimental environments were intended to model actual battlefield environments. The simulated tasks were primarily detection and discrimination tasks, although other inference problems could also be tested in the same environments. The environments consisted of a discrete number of possible target locations. Targets could exist in any of the locations and could be of several types. *A priori* probabilities for each target type at each possible location were known by the sensors. The specifications of the environments for each simulated experiment are given in Chapter 4.

1.4.2 Simulated Sensors

Sensors were simulated to have a variety of abilities and limitations. Both mobile and stationary, low and high power, single and multi-modal sensors were used in the simulations. The sensor differences resulted in varying abilities regarding resolution, noise suppression, and measurement time and range. The simulated sensor configuration, including choice of sensing location and modality, affected the sensors' probability of false alarms or missed detections. This is true for real sensors. For instance, if the magnetometer from [10] is deployed to a locality high in metallic concentration, the probability of false alarm will increase dramatically. Similarly, if the airborne, bimodal SAR discussed in [30] operates in HRR mode over a forest, the probability of missed detection will be increased due to foliage occlusion. This sensor/environment dependence was simulated by assigning to each environment locality and each sensor and mode of operation a Signal-to-Noise ratio (SNR). High SNRs result in better ability to detect targets. The specific sensor model is given in Chapter 4.

1.5 Related Work

1.5.1 Scheduling in a Sensor Network

A starting point for this Master's Thesis is Maurice Chu's doctoral thesis [10]. In this work, Dr. Chu develops an optimal method for distributed inference in a large scale sensor network. Additionally, he proposes a technique for sub-optimal inference when only local knowledge is available, as when only a subset of the network sensors report measurments. This may be planned (e.g. to limit power consumption), or unplanned (e.g. sensor failure, network jamming, message loss). He uses a combinatorial approach to sensor planning, creating a complete set of sensor plans for all likely sensor readings. He then derives a system of triggering events. The main purpose of this system is to limit the amount of unnecessary information gathered in the sensor network. Evaluation of these triggers can be distributed among network nodes, minimizing the time necessary for detection. Foundational work for Dr. Chu's method includes [9, 43] where the Information Driven Sensor Query (IDSQ) method for sensor scheduling was derived. IDSQ has been critiqued and expanded upon in [13, 41, 22].

1.5.2 Sensor Network Resource Management

A central focus of the thesis is how to extend the information theoretic principles derived in [9, 10] to a situation in which multiple sensors can be simultaneously active. In [42] many of the issues involved in multi-sensor management are surveyed, including issues related to the optimal placement of sensors within an area of interest, the benefits and constraints of decentralized control, and sensor cooperation.

In addition to inter-sensor management, the optimal choice of mode for a single sensor is also relevant to the development. In [30] an optimization involving a single, multi-modal sensor suite was presented. Additionally the issue of sensor placement for mobile, active sensors described in [11, 16] can be considered as attempts to optimize over a continuous mode of operation (in this case, geographical placement) in order to achieve goals which may change with time. The ability to optimize over continuous modes is essential to planning in networks involving active sensors.

1.5.3 General Optimization

Applicable research not specific to sensors and sensor networks includes the theory of dynamic programming (cf. [5]) and the closely related work in Markov Decision Processes [12]. The basic theory of dynamic programming extends traditional optimization analysis to a set of problems which vary with time. It provides a structure for decision feedback and sequential decision analysis, enabling information to be incorporated in an optimal manner while it is being gathered. This applies directly to the thesis, in which a sensor plan may need to be modified as new sensor measurements are made and incorporated into the likelihood function.

Dynamic programming is a powerful tool, but can be computationaly prohibitive to implement, particularly in problems where the state space is large. To address this issue, learning algorithms have been developed to approximate the optimal solutions found through dynamic programming. Neurodyanmic programming [6], or "reinforcement learning" as it is sometimes known, uses neural network and other approximation architectures to adapt dynamic programming methods to complex environments.

Dynamic programming has been previously applied to sensor networks in [39, 31]. It has also been used to analyze single sensor management [7] as well as to general real-time environments [2].

In addition to dynamic programming, other optimization techniques may be employed, particularly in optimizing the cost-to-go subproblem. These techniques may include nonlinear, mixed-integer, and/or constrained programming [4, 21].

1.6 Contributions

Previous research in sensor scheduling has focused primarily on improving the quality of inference [9, 13, 39, 10]. When resource costs have been considered they have been limited to power consumption due to communication [22, 41], and have generally been treated as constraints on the set of feasible solutions rather than as elements of the objective. This thesis formulates the problem as one in which the quality of the inference is the constraint, while the system cost is a function of the amount of time the schedule takes. The primary objective is not to achieve a high quality of inference independent of how long it takes, but rather to achieve a "good enough" estimate as quickly as possible. Time minimization is fundamentally important in many applications, and examining this formulation will expand the usefulness of sensor networks in both military and industrial applications.

Additionally, the minimum time formulation necessitates a consideration of sensor interactions. In much of the previous work in sensor networks, it is assumed that a single "leader" sensor is active at any given time and all other sensors in the network are inactive. This assumption simplifies much of the analysis, but in the formulation for this thesis it significantly reduces the optimality of the solution. It is essential to consider methods for sensor scheduling when multiple sensors may be active at the same time. Applying elements of game theory to this problem of multiple sensor coordination is an important contribution of this thesis.

One of the central issues facing the United States military is the efficient use of multiple sensor platforms. The past twenty years have seen an unprecedented number of sensors developed for military application. There are sensors in the air, on land, on sea, underwater and even in space. Each sensor has unique capabilities that make it more valuable in some situations and less valuable in others. The wealth of sensors available presents two significant problems: one, how to deal with the copious amounts of raw data produced (so much that it often overwhelms human analysts); and two, how to enable efficient sensor cooperation. Since the sensors were developed individually they often suffer from what's called stove-pipe sensing, meaning it is difficult to use sensors cooperatively.

The principles derived in this thesis, especially the cooperative architecture developed in Chapter 3, address these two problems. Through increasing the autonomy of the sensor network, less data need be analyzed by humans. The architecture also provides a framework in which several sensors can efficiently cooperate in a highly dynamic and unknown environment. The contributions of this thesis will directly and immediately influence the future development of integrated sensing platforms for the U.S. military through ongoing research at MIT Lincoln Laboratory in the Integrated Sensing and Decision Support (ISDS) group.

1.7 Thesis Organization

This chapter has served as an introduction to some of the issues involved in sceduling sensor networks. Specifically, a cannonical detection scenario based on battlefield awareness has been presented, an objective of minimizing time to reach an acceptable state of certainty has been stated, and a method for accomplishing the objective has been proposed.

In Chapter 2, sensor network inference problems will be discussed in greater detail. Then the minimum time optimization formulation from Section 1.3.3 will be developed and analyzed using methods from dynamic programming.

Chapter 3 will discuss the need for coordination in solving certain inference problems in sensor networks. It will then present a cooperative, distributed method for achieving the optimal sensor deployments derived in Chapter 2.

Chapter 4 will introduce simulations used to verify the algorithm proposed in Chapter 3. The simulation will consist of general classes of environments and a suite of available sensors. Results from the simulation will be presented along with an analysis of the observed strengths and limitations of the algorithm.

Finally, in Chapter 5 conclusions will be drawn as to the viability of the derived algorithm and its possible extension to new domains.

Chapter 2

Dynamic Problem Formulation

2.1 Sensor Network Model

As discussed in Chapter 1, an important use of sensors is in improving battlefield awareness. Sensors currently used in this type of situation are highly varied and include both active and passive, mobile and stationary, single and multi-modal sensors. Figure 2-1 depicts a typical battlefield scenario with a network made up of highly varied sensors.

Each sensor in a network has a set of possible measurements it may receive. For instance, the stationary magenetometer described in [10] measures either a 0 or a 1, depending on whether any magnetic material is detectable in its area of observation. A mobile radar has a different set of possible measurements, based in part on its position, velocity, pulse frequency, dwell time and other operational paramaters [34]. Using such varied assets conjointly is made possible if the network is capable of aggregating the various measurements into a single, universally understood model.

Consider sensor $s_i \in S$, where S is set of all sensors in the network. Assume s_i has a **measurement model** consisting of the 2-ple $(Z_i, p_i(z|x))$, where Z_i is a set of possible measurements, and $p_i(z|x)$ is the probability of observing measurement $z \in Z_i$, conditioned on the true state x. Furthermore, assume that for all s_i the set of possible true states is fixed and known and that measurements are conditionally independent given the true state, meaning that $\forall\ i, j\ p_{ij}(z_1, z_2|x) = p_i(z_1|x) * p_j(z_2|x),\ \ z_1 \in Z_i, z_2 \in Z_j$.

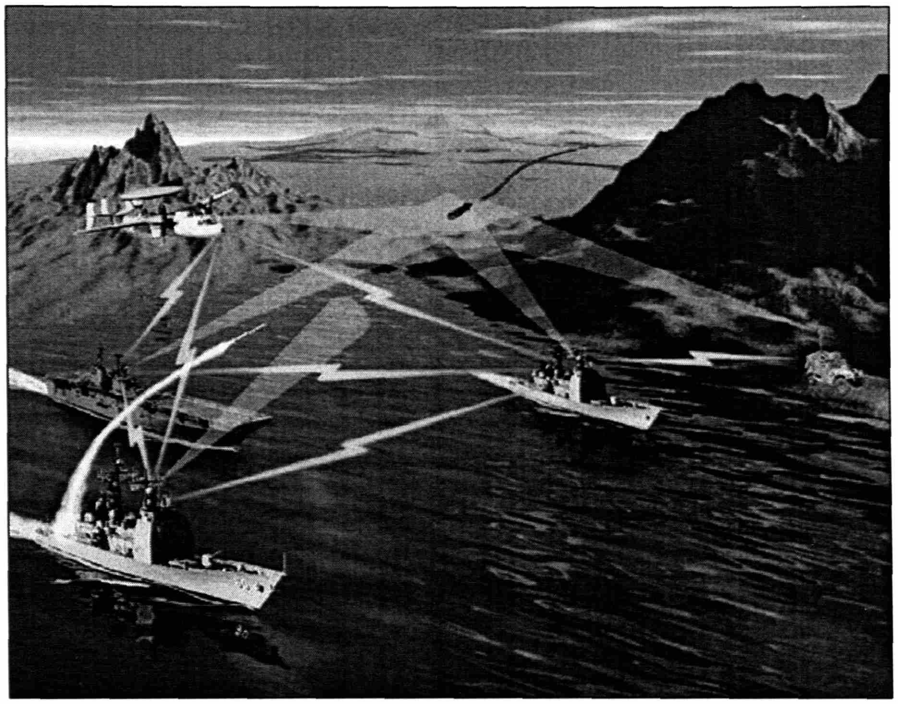

Figure 2-1: Sensor network in a naval battlefield scenario (courtesy of [26])

Under these assumptions the network can maintain a global probability function $\pi(x|\mathbf{z})$ (where $\mathbf{z} = \{z_i\}_{i=1}^{N}$) if each time sensor s_i obtains a new measurement, the global probability function $\pi(x|\mathbf{z})$ is updated by the sensor to be the optimal posterior probability function. For detection problems this would be accomplished by s_i calculating and then broadcasting to the rest of the network the Bayesian update

$$\pi(x|\{z_n\}_{n=1}^{N+1}) = \frac{\pi(x|\{z_n\}_{n=1}^{N}) * \pi_i(z_{N+1}|x)}{\sum_{x\in X} \pi(x|\{z_n\}_{n=1}^{N}) * \pi_i(z_{N+1}|x)} \tag{2.1}$$

where z_{N+1} is the most recent measurement observed by s_i. In other inference situations (e.g. tracking, localization, discrimination) other update rules might be used, but the process would be the same. The sensor taking the measurement updates the (global) posterior probability function using its (local) measurement model and the known update rule $f(\pi, z, p(z|x))$. In general, the posterior distribution will be updated after the N^{th} measurement by $\pi(x|\{z\}_{n=1}^{N}) = f(\pi(x|\{z\}_{n=1}^{N-1}), z_N, p(z_N|x))$.

An inference problem is defined by the 3-ple $(X, p_0(x), f(\cdot,\cdot,\cdot))$. X is the set of possible decisions, corresponding to the fixed and known true states of the sensor models, $p_0(x)$ is the initial probability of each of the states and $f(\cdot,\cdot,\cdot)$ is the update rule. An inference problem is **solved** by a decision rule $D : [(X, p_0(x), f(\cdot,\cdot,\cdot)), \mathbf{z}] \mapsto X$. Thus, an inference problem is solved when one of the possible states is decided for and the rest are decided against.

2.2 Optimal Formulations of Inference Problems

As stated in Chapter 1, an inference problem is any instance of converting raw data to information. In the case of sensor networks, inference problems include transforming a set of RFID tag responses to locations, determining the location of possible victims from search and rescue robots' sensor responses, or localizing, tracking and identifying an enemy target within a battlefield. Inference problems are as diverse as the networks used to solve them, but all inference problems involve the ability to extract information from raw sensor data.

A sensor network's capability to solve inference problems can be formalized as an optimization problem. The most general format of an optimization problem is

$$\begin{aligned} x^* &= \arg\min_{x \in X} f(x) \\ & \quad \text{s.t. } g(x) \in G \end{aligned} \tag{2.2}$$

The function $f(x)$ is called the objective function, $g(x)$ is called the constraint function or constraint and G is the constraint set. A solution x is called **feasible** if $g(x) \in G$, and the set $F = \{x : g(x) \in G\}$ is called the **feasible set**. A solution x^* is called **optimal** if it is feasible and $f(x^*) \leq f(x)\ \forall x \in F$.

In sensor networks, there is some variability in how the objectives and constraints from Equation 2.2 are chosen. This variability results in a variety of optimal formulations, each providing unique insight into the operation of the sensor network. Which formulation should be used is dependent on the specific goals and design of the system.

2.2.1 Objectives and Constraints

As discussed in Section 1.1.1, there are several possible costs due to resource consumption in a sensor network, including costs due to communication, time, and energy. We will refer to these collectively as **consumption costs**. Additionally, inaccurate solutions to the inference problem (e.g. missed detections, bad localizations, etc.) carry some cost. We will term these **quality costs**. The objective function and the constraints for Equation 2.2 derive from combining quality and consumption costs. Different categorizations lead to different optimization formulations.

Minimum Consumption Formulation

A sensor network must consume some amount of its resources while solving an inference problem, meaning it must use some amount of time, energy, communication and so forth. In the minimum consumption formulation, the objective function is a measure of the resource consumption of a network. The quality costs are incorporated through the constraint function. The goal under this formulation is to consume as little as possible of the network resources under the constraint that the resulting inference problem solution meet some minimum quality requirement. As an example of a minimum consumption formulation, consider the following problem statement: minimize the energy used in a network while guaranteeing a probability of detection no less than γ. In this case the objective function is the energy consumed by the network (a consumption cost), the constraint function is the probability of detection (quality cost) and the constraint set is the set of all solutions with probability of detection greater than or equal to γ.

One of the challenges of using a minimum consumption formulation for systems with diverse consumption costs is in the determination of an objective function that balances the consumption of the various resources. For instance, which solution has lower cost: one that satisfies the constraint in 5 minutes using 1 W of power, or one that satisfies the constraint in 50 minutes using 1 mW of power? The answer depends inherently on the purpose and design of the network. Aggregating consumption costs into a single objective function can be very challenging.

Maximum Quality Formulation

In this case, the objective is to minimize the quality costs of the solution. Quality of the solution refers to the probability of making a wrong decision. Take for example a localization problem. The inference problem is to determine the location of a target. The objective function might be the expected mean square error of the estimated location from the true location. The objective function is a measure of the quality of the inference problem solution.

In the maximum quality formulation the consumption costs are incorporated through constraint functions. Thus, using the localization example, one constraint might be that total energy expenditure in the network is less than E mW, or that a final estimate is reached in no more than T seconds. This formulation is particularly appropriate in systems with strict resource constraints, such as sensor networks with energy lifetime limitations or hard time constraints. It has the advantage of never comparing minutes to milliwatts, because each resource can be constrained separately. Thus we can constrain the total time to be less than T, the total energy used to be less than E and so forth.

Hybrid Formulations

In some situations it may be beneficial to incorporate both consumption and quality costs into the objective function and solve an unconstrained optimization problem. In this case, the problem of disparate costs is increased, because the quality cost must be explicityly traded off against costs measured in seconds or communication hops. Aggregating these into a single objective function usually requires hand-tweaking paramaters by an involved human, who makes the decision about how valuable each factor is in determining "optimality."

It is also possible to have a mixed hybrid equation in which some subset of the consumption and quality costs are considered constraints, while the rest are aggregated into the objective function. For every inference problem there are multiple possible formulations of the optimization equation, each with a different definition of what objective is being optimized and over what set of constraints.

2.2.2 Actions in Sensor Networks

A sensor's primary purpose it to gain information about its environment. This raw sensor data is then processed in order to extract information about the true state of the environment. This process of gathering data and extracting information is the inference problem referred to in Section 2.1. In this model there is no specification of how (i.e. in what order, under what parameters, etc.) sensors obtain measurements. The determination of these quantities is the primary goal of optimization in the sensor network.

Sensor actions may be defined differently depending on the network model. For instance, in [35] sensor actions include taking a measurement, entering sleep mode, aggregating several measurements, and sending a message to other sensors in the network. In [9, 41, 13] sensor actions consist only of choosing the next "leader" for the network. In [30] the choice is of which mode of the sensor to operate under while taking measurements. Each of these choices for the set of actions in a sensor network may be appropriate depending on the network under consideration. In this thesis we combine the latter two models and consider the actions in a sensor network to be choosing 1) the next "leader" for the network and 2) the mode in which the "leader" should take its next measurement. In Chapter 3 we will revisit the limitation of choosing a specific "leader" and consider models where several sensors may make measurements simultaneously.

By the formulation above it can be seen that an action in a sensor network can be represented as (s_i, m) where $s_i \in S$ and $m \in M_i$ where M_i is the set of all possible modalities under which s_i can take a measurement, or operate. An element of the set M_i is sometimes called a vector of operational parameters [33]. In a heterogeneous sensor network each sensor may have a unique M_i. It may include both discrete and continuous elements. For instance some radars have the capability of operating in either GMTI or HRR modes. These are discrete modes. When operating in GMTI, a radar may have the capability of operating in a range of pulse frequencies. This is an example of a continuous mode. For sensors with several possible operating modes, the set M_i will be a complicated, nested data structure representing all of the valid combinations of operating parameters. For mobile sensors, as well as static sensors with steerable capabilities, one of the important operating parameters is which physical locations to survey when taking measurements. The choice of this location parameter is central to much of the development of this thesis.

Using the concept of modes we can enhance the measurement model given in Section 2.1. Recall that the measurement model for s_i was $(Z_i, p_i(z|x))$. Consider now that the set of possible measurements in one mode of operation may not be the same as the set of measurements in a different mode. Define $Z_i(m)$ to be the set of possible measurements for s_i under mode $m \in M_i$ and $Z_i = \bigcup_{m \in M_i} Z_i(m)$. This definition partitions the measurement space among the several modes of sensor operation. Also, $p_i(z|x)$ will depend on m which we will denote $p_i(z|x;m)$ meaning the probability of sensor s_i deployed in mode m observing measurement z when the true state is x.

2.3 Minimum Time Equation

In much of the liturature dealing with inference problems in sensor networks (e.g. [9, 13, 27]), the optimization formulation used has been the maximum quality formulation described in Section 2.2. This formulation is appropriate when considering a network in which there are hard constraints on consumption (such as the energy lifetime of low-power sensor networks). Hybrid formulations have also been analyzed in [41, 35] in which consumption costs were considered as part of the objective in conjunction with quality costs. However to the knowledge of the author, at this date no study has been made of networks with a primary goal of minimum time inference. Under the categorization from Section 2.2 this is a minimum consumption formulation in which the objective function is time and the constraint function is a measure of the quality of the solution of the inference problem.

There are many possible measures of quality of the solution, including (but certainly not limited to) the Shannon entropy, relative entropy, probability of detection or mean-square localization error. In general, quality costs or measures will be represented by $\mu(D)$ where D is a decision rule as defined in Section 2.1. Frequently the quality costs depend only on the posterior probability function, rather than the complete decision rule. For simplicity, such measures will be denoted $\mu(\pi(x|\mathbf{z}))$ where $\pi(x|\mathbf{z})$ is the posterior probability function.

Some quality measures such as entropy or localization error are inversely related to quality (i.e. quality goes up as the measure goes down). Others, such as probability of detection are directly related to quality. If the measure $\mu(\pi(x|\mathbf{z})$ is of the former type, then a typical quality constraint would be $\mu(\pi(x|\mathbf{z})) \leq \gamma$. For measures of the latter type, the constraint would be $\mu(\pi(x|\mathbf{z})) > \gamma$. In general we will consider measures of the former type, specifically entropy measures such as the Kullback-Liebler distance between the posterior density and the density representing the true state.

2.3.1 Dynamic Program Formulation

As discussed in section 1.3.3, inference problems for real sensor networks are stochastic optimization problems and can therefore benefit from dynamic analysis. Specifically, breaking the problem into a sequence of actions followed by observations and allowing future actions to depend on previous observations allows for better solutions. A diagram of the organization of a dynamic program is shown in Figure 2-2. In the figure, a_i is the i^{th} action taken, d_i is the i^{th} decision and z_i is the i^{th} sensor measurement received.

Our dynamic optimization equation is then

$$J(\pi_t) = \min_{a \in A_t} \left(E[t(a)] + \sum_{z \in Z_i(m)} p(z) * J(f(\pi_t, z, p_i(z|x)))] \right) \tag{2.3}$$

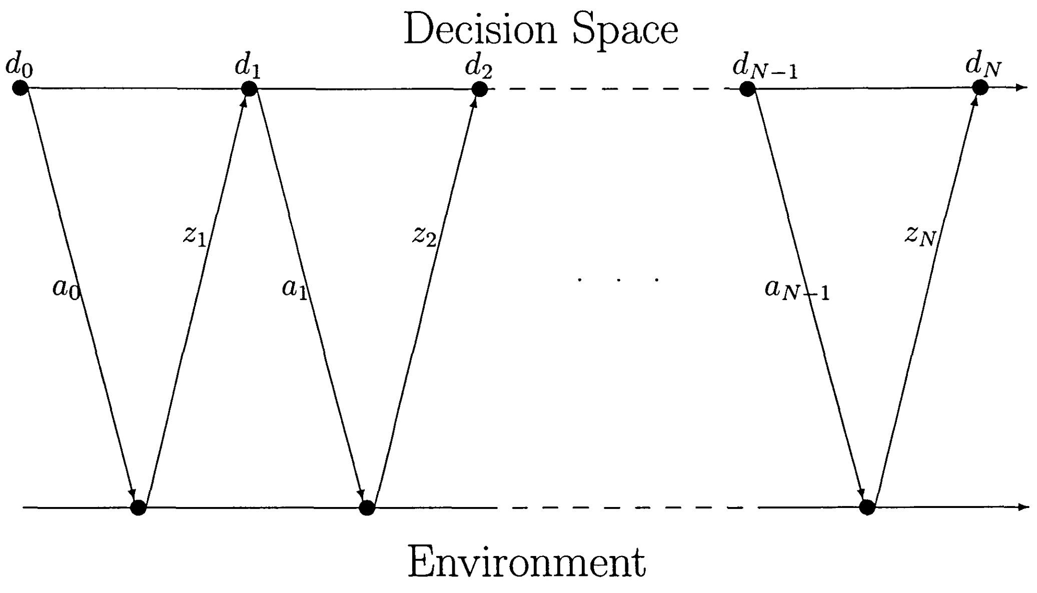

Figure 2-2: Dynamic decision flow

with $J(\pi_t) = 0$ if $\mu(\pi_t) \leq \gamma$, and where π_t is the posterior density at time t, $t(a)$ is the time it takes to perform action a, $p(z) = \sum_{x \in X} p(x) * p_i(z|x)$ and $f(\pi, z, p(z|x))$ is the state update function defined by the inference problem (see Section 2.1).

2.4 Solution to the Minimum Time Formulation

In most cases, solving dynamic programs like that represented in Equation 2.3 requires an insupportable amount of computation, necessitating the adoption of approximate methods, such as heuristics. However, under certain assumptions optimal solutions can be determined *a priori*. In Section 2.4.1 a sensor network in which all measurements are predetermined is examined. It is shown that if only one sensor can be queried or deployed at a time and when $\mu(\pi_t) - \gamma$ is large relative to $E_z[\mu(\pi_t) - \mu(f(\pi_t, z, p_i(z|x)))]$, the optimal action is the one which maximizes the expected rate of decrease of $\mu(\pi_t)$.

Then, in Section 2.4.2, the solution is extended to a limited number of stochastic sensor networks, in which measurements are not predetermined. By analyzing this limited set of (frankly, unrealistic) sensor networks, it is hoped that a heuristic can be developed for the general problem of time optimal inference in a sensor network.

2.4.1 Optimal Actions in Deterministic Networks

Consider a deterministic sensor network. For each sensor and each mode of operation (s_i, m), $s \in S, m \in M_i$ there is a predetermined measurement z. This corresponds to a network where

$$p_i(z|x;m) = \begin{cases} 1 & \text{if } z \in Z_i(m) \\ 0 & \text{otherwise} \end{cases}$$

Furthermore, let the state update equation $f(\pi_t, z, p_i(z|x;m)$ be such that

$$\mu(\pi_t) - \mu(f(\pi_t, z, p_i(z|x;m))) = C(s_i, m)$$

Thus, for a given sensor and mode of operation the quality measure decreases by a constant amount.

As before let an action a be the 2-ple (s_i, m) and consider for the moment only discrete sensor modes. The set of all possible actions for the sensor network is $A = \bigcup_{s_i \in S} \{(s_i, m)\}_{m \in M_i}$ and the total number of possible actions is $|A| = \sum_{i=1}^{|S|} |M_i| = N$. For each action $a_i = (s_j, m)$ denote $\mu_i = C(s_j, m)$ and denote the time necessary for action a_i as t_i.

Equation 2.3 can be reformulated as

$$J(\pi_t) = \min_{i \in 1:N} (t_i + J(f(\pi_t, a_i))) \tag{2.4}$$

where $J(\pi_t) = 0$ if $\mu(\pi_t) \leq \gamma$ and expectations and the dependance of the state update equation on $p_i(z|x;m)$ have been dropped because of the deterministic assumption.

A **pure strategy**, u_i, is one in which action a_i is used exclusively. The cost associated with a pure strategy is denoted $J_{u_i}(\pi_t)$. Let $J^*(\pi)$ be the optimal cost function for probability π. The maximal information acquisition rate principle states that, under certain conditions, an optimal action is the one with maximium information acquisition rate (IAR), where the IAR is defined as $\rho_i = \frac{\mu_i}{t_i}$

Proposition 2.1 *For large enough T, if $J^*(\pi) > T$,*

$$i^* = \arg\max_{i \in N} \frac{\mu_i}{t_i}$$

Proof: Assume, WLOG that actions are ordered by

$$\frac{\mu_1}{t_1} > \frac{\mu_2}{t_2} > \cdots > \frac{\mu_N}{t_N} \tag{2.5}$$

Also, let $D = (\mu(\pi) - \gamma)$.

Consider the (non-existant) instantaneous action $\tilde{a}_2$ with IAR $\tilde{\rho}_2 = \frac{\mu_2}{t_2}$ and its associated pure strategy $\tilde{u}_2$. The cost of strategy $\tilde{u}_2$ is

$$J_{\tilde{u}_2}(\pi) = \frac{D}{\mu_2} t_2 \tag{2.6}$$

Let $J^*_{[2:N]}(\pi)$ be the cost associated with the optimal strategy among all strategies that don't include action a_1. I will first show that

$$J_{\tilde{u}_2}(\pi) < J^*_{[2:N]}(\pi) \tag{2.7}$$

Then I will show that, for D large enough,

$$J_{u_1}(\pi) \leq J_{\tilde{u}_2}(\pi) \tag{2.8}$$

Proof of Equation 2.7

I will prove Equation 2.7 inductively on the number of times and action other than $\tilde{a}_2$ is taken. For the base case consider strategy σ_1 where action $a_i, i \in [2:N]$ is taken once and action $\tilde{a}_2$ is taken otherwise.

$$J_{\sigma_1}(\pi) = \frac{D - \mu_i}{\rho_2} + t_i = \frac{D}{\mu_2} t_2 - \frac{\mu_i}{\mu_2} t_2 + t_i = J_{\tilde{u}_2}(\pi) + t_i - \frac{\mu_i}{\mu_2} t_2 \tag{2.9}$$

where the last step was accomplished by substitution from Equation 2.6. Now, by 2.5

$$\frac{\mu_i}{t_i} > \frac{\mu_2}{t_2} \Rightarrow t_i > \frac{\mu_i}{\mu_2} t_2 \tag{2.10}$$

Combining Equations 2.9 and 2.10 we get

$$J_{\sigma_1}(\pi) > J_{\tilde{u}_2}(\pi) \tag{2.11}$$

For the induction step, consider strategies σ_k and σ_{k+1}. Under σ_k, k actions other than $\tilde{a}_2$ are taken. By an argument similar to the base case

$$J_{\sigma_{k+1}}(\pi) = J_{\sigma_k}(\pi) + t_i - \frac{\mu_i}{\mu_2} t_2$$

Then, from Equation 2.10 it's obvious that

$$J_{\sigma_{k+1}}(\pi) > J_{\sigma_k}(\pi)$$

Thus

$$J_{\sigma_k}(\pi) > J_{\tilde{u}_2}(\pi) \ \forall \ k$$

For all π, the optimal strategy among all strategies that don't include a_1 is in σ_k for some k. Therefore

$$J_{\tilde{u}_2}(\pi) < J^*_{[2:N]}(\pi)$$

Proof of Equation 2.8

Repeating Equation 2.6

$$J_{\tilde{u}_2}(\pi) = \frac{D}{\mu_2} t_2 = \frac{D}{\rho_2}$$

Consider the pure strategy u_1 in which only action a_1 is taken. The cost for strategy u_1 is

$$J_{u_1}(\pi) = \lceil \frac{D}{\mu_1} \rceil t_1 \tag{2.12}$$

By simple algebra we have

$$J_{u_1}(\pi) \leq \frac{D}{\mu_1}t_1 + t_1 = \frac{D}{\rho_1} + t_1 \tag{2.13}$$

If π is such that

$$D \geq \frac{t_1}{\frac{1}{\rho_2} - \frac{1}{\rho_1}} \Rightarrow \frac{D}{\rho_2} \geq \frac{D}{\rho_1} + t_1$$

then, from Equations 2.6 and 2.13

$$J_{u_1}(\pi) \leq J_{\tilde{u}_2}(\pi) \tag{2.14}$$

Together, Equations 2.7 and 2.8 imply that, if $\mu(\pi) - \gamma$ is large with respect to the inverse difference of the inverse of the two greatest IARs, then the optimal action sequence will include the action with maximum IAR. Furthermore, since all results of actions are deterministic, action order is superflous. Therefore if

$$i = \arg \max_{i \in [1:N]} \rho_i$$

then action a_i is an optimal action for the network.

The maximum network IAR is what is called a dynamic allocation index, or a Gittens index [15]. Gittens' indices are solutions to a wide range of dynamic problems in which time is a factor. It should be noted that the inference problems in sensor networks do not generally satisfy the requirements for an optimal Gittens index solution. Particularly, if the sensor model for the network is not known, but is being discovered, then the index solution dervied above may be significantly suboptimal.

2.4.2 Optimal Actions in Stochastic Networks

We can extend the above analysis to include cases in which both the time required to take each action, t_i, and the subsequent decrease in uncertainty for the inference problem, μ_i, are random variables. Making these values random variables allows for a more realistic representation of a sensor network. Results similar to those from

Section 2.4.1 can be derived as long as the random variables are assumed independent. First, the results will be extended to stationary systems where the expected values of the random variables remain constant, then to a set of interesting, non-stationary systems.

Stationary Systems

In the case of stochastic systems with constant mean, the cost function 2.3 can be written as

$$J(\pi) = \min_{i \in 1:N} \left(E[t_i + J(f(\pi, a_i))]\right) \tag{2.15}$$

where $J(\pi) = 0$ if $\mu(\pi) \leq \gamma$. This is exactly equivalent to Equation 2.4 except that the cost is now taken under an expectation on both the immediate time cost of action i as well as on the future cost-to-go.

Proposition 2.2 *For large enough T, if $J^*(\pi) > T$,*

$$i^* = \arg\max_{i \in N} E[\frac{\mu_i}{t_i}]$$

Proof: Since $E[\mu_i]$ and $E[t_i]$ are assumed independent the cost of pure strategy u_i is

$$J_{u_i}(\pi) = \lceil \frac{D}{E[\mu_i]} \rceil E[t_i] \tag{2.16}$$

$J_{u_i}(\pi)$ can be bounded by

$$J_{u_i}(\pi) \leq \frac{D}{E[\rho_i]} + E[t_i]$$

where $\rho_i = \frac{\mu_i}{t_i}$. Again, assuming Equation 2.5 and assuming that

$$D \geq \frac{E[t_1]}{\frac{1}{E[\rho_2]} - \frac{1}{E[\rho_1]}} \Rightarrow \frac{D}{E[\rho_2]} \geq \frac{D}{E[\rho_1]} + E[t_1]$$

we have the result

$$J_{u_1}(\pi) \leq \frac{D}{E[\rho_1]} + E[t_1] \leq \frac{D}{E[\rho_2]} = J_{\tilde{u}_2}(\pi) \tag{2.17}$$

where $\tilde{u}_2$ is the instantaneous pure strategy with IAR equal to ρ_2. Consider the space of all strategies that only include actions $a_{[2:N]}$ as well as action $\tilde{a}_2$. The cost for strategy σ from that space is

$$J_\sigma(\pi) = \sum_{i=1}^{N} \sigma_i E[t_i] = \sum_{i=1}^{N} \frac{D_i}{\rho_i} \tag{2.18}$$

where σ_i represents the number of times a_i is taken (to simplify math, σ_1 is assumed to be the number of times action $\tilde{a}_2$ is taken) and $D_i = E[\mu_i] * \sigma_i$. Since $\rho_2 > \rho_3 > \cdots > \rho_N$

$$J_\sigma(\pi) = \sum_{i=1}^{N} \frac{D_i}{\rho_i} > \sum_{i=1}^{N} \frac{D_i}{\rho_2} = J_{\tilde{u}_2}(\pi) \text{ if } D = \sum_{i=1}^{N} D_i \tag{2.19}$$

Furthermore, since action $\tilde{a}_2$ has instantaneous cost

$$\forall\, \pi\ \exists\, \sigma \text{ s.t. } D = (\mu(\pi) - \gamma) = \sum_{i=1}^{N} D_i$$

Equations 2.17 and 2.19 together imply that, for D large enough and for constant and independent $E[\mu_i], E[t_i]\ \forall i$, an optimal action is the one that maximizes $E[\rho_i]$.

Non-stationary Systems

Sensor systems are, in general, non-stationary. If sensor s_i is queried once, the expected decrease in uncertainty is μ_i. If sensor s_i is queried again with the same parameters the expected uncertainty decrease will not generally be μ_i. To more acurately model the decreasing utility of actions, the following non-stationarity assumption will be made.

Assumption 2.1 *Each time action a_i is taken, the corresponding expectation of decrease in uncertainty, $E[\mu_i]$ contracts by α. All other random variables remain stationary.*

The dynamic program is still as in 2.15

$$J(\pi) = \min_{i \in 1:N} \left(E[t_i + J(f(\pi, a_i))] \right)$$

but the expected decrease in uncertainty is no longer constant.

Let σ be a vector such that σ_i is the number of times a_i is taken. Also, denote the initial expected decrease in uncertainty due to action a_i as $E[\mu_i]$. If the initial uncertainty of the system is $\mu(\pi)$, the expected uncertainty after taking the actions specified by σ is

$$\begin{aligned} E[\mu(\pi_\sigma)] &= \mu(\pi) - \sum_{i=1}^{N} (E[\mu_i] \sum_{j=0}^{\sigma_i - 1} \alpha^j) && (2.20) \\ &= \mu(\pi) - \sum_{i=1}^{N} (E[\mu_i] * \frac{1-\alpha^{\sigma_i}}{1-\alpha}) && (2.21) \end{aligned}$$

The expected time under strategy σ can be written as

$$J_\sigma(\pi) = \sum_{i=1}^{N} \sigma_i E[t_i]$$

with the understanding that $E[\mu(\pi_\sigma)] \leq \gamma]$. The optimal cost can be written

$$J^*(\pi) = \min_{\sigma \in \Sigma} J_\sigma(\pi)$$

which can, in turn be rewritten as the program

$$\min_{\sigma \in \Sigma} \sigma^{\mathrm{T}} E[t]$$

$$\text{s.t.} \sum_{i=1}^{N} (E[\mu_i] * \frac{1-\alpha^{\sigma_i}}{1-\alpha}) \geq D$$

where $D = \mu(\pi) - \gamma$. Since α is constant, the constraint can be rewritten as

$$\begin{aligned} \sum_{i=1}^{N} (E[\mu_i] * (1-\alpha^{\sigma_i})) &\geq D(1-\alpha) \\ \sum_{i=1}^{N} (E[\mu_i] - E[\mu_i]\alpha^{\sigma_i})) &\geq D(1-\alpha) \end{aligned}$$

The constraint is non-linear in σ. Consider instead the linearized system, where the α^{σ_i} terms are linearized about the nominal point $\sigma_i^0 = 0$. The resulting constraint is

$$\begin{aligned}
\sum_{i=1}^{N}(E[\mu_i] - E[\mu_i](\ln\alpha)\sigma_i &\geq D(1-\alpha) && (2.22)\\
-(\ln\alpha) * \sum_{i=1}^{N} E[\mu_i](\sigma_i) &\geq D(1-\alpha) - \sum_{i=1}^{N} E[\mu_i] && (2.23)\\
\sum_{i=1}^{N} E[\mu_i](\sigma_i) &\geq \frac{1}{\ln\alpha}(\sum_{i=1}^{N} E[\mu_i] - D(1-\alpha)) && (2.24)
\end{aligned}$$

where, in the final step, the inequality doesn't change directions since $-\ln\alpha > 0$. The constraint is now the same as in the stationary case, except that strategy σ must satisfy Equation 2.24 rather than $\sum_{i=1}^{N} E[\mu_i](\sigma_i) \geq D$. Letting

$$\tilde{D} = \frac{1}{\ln\alpha}(\sum_{i=1}^{N} E[\mu_i] - D(1-\alpha))$$

the analysis for the stationary case demonstrates that an optimal action for the linearized system is the one that maximizes the expected IAR. Unlike the systems analyzed previously, this may not always be the same action, due to the decreasing value of $E[\mu_i]$.

One caveat must be made in non-stationary systems that follow the model described above. Due to the asymptotic properties of the expected decreases in uncertainty, it is possible that the set of feasible solutions to the dynamic program is empty. If $\sum_{i=1}^{N} E[\mu_i](\frac{1}{1-\alpha}) \leq D$ there is no action sequence that leads to a solution of sufficient expected certainty. This can be addressed by adding the system constraint that

$$D(1-\alpha) \geq \sum_{i=1}^{N} E[\mu_i]$$

This constraint will also ensure that $\tilde{D} > 0$.

2.5 Conclusions

We have shown that in specific types of sensor networks when the distance between the current uncertainty and the threshold is large, the optimal action is the one that maximizes the rate of information acquisiton. This can be compared to the results in [41] where IDSQ is extended to include communication costs. While the above discussion focused on costs due to time, it could be easily extended to situations where the main cost is communication as in the networks considered in [22, 41]. In Chapter 4 the IDSQ algorithm will be compared to the above formulation through simulation.

The maximum IAR principle is particularly well-suited to sensor network applications because it can be implemented in a distributed fashion. Because the optimal solution is the one with maximum IAR, if the assumption is made that a sensor's IAR is independent of other sensors, each sensor can calculate its own optimal IAR among all its possible modes of operation. A distributed leader election algorithm [23] can then be used to nominate the sensor with maximum individual IAR. Finding the maximum IAR for a single sensor will generally involve a non-linear optimization problem. Interior point methods [4] and randomized solution methods may need to be used in order to find the operating mode with maximum IAR. Due to the complexity of the solution methods, finding the utility of the maximum IAR method may be limited to sensors with significant computational ability.

One assumption made in the much of the sensor network literature (e.g. [9, 27, 41]) as well as in the preceeding derivation of the maximum IAR principle, is that only one sensor, called the leader, is active in a network at any given time. In networks where the primary goal is to approximately solve an inference problme in minimum time this assumption may be prohibatively limiting. Chapter 3 will show how to extend the principle of maximum IAR to networks in which several sensors can operate simultaneously in order to cooperatively solve the given inference problem.

Chapter 3

Cooperation for Time Optimal Inference

3.1 Introduction

In Chapter 2 problems of inference in a sensor network were formulated as a dynamic program. Specifically, a minimum time optimization problem was formulated as a dynamic program. The program was then solved for single sensor deployments in both deterministic and stochastic networks.

The simple index rule solution presented in Chapter 2 can be complicated in networks where sensors interact. When two sensors operate simultaneously in the same area there is the possibility of sensor interaction. This interaction may be constructive or destructive.

Example 1: Constructive Interaction Consider three ground vehicles equipped with video sensors and tasked with locating a target of interest. By coordinating their sensing, the three sensors are able to triangulate the target location with a high degree of certainty. But when any one sensor doesn't participate in the sensing, triangulation is no longer possible and the localization is much less accurate. The three sensors were able to use simultaneous sensing to achieve a level of certainty greater than any of the three acting singly.

Example 2: Destructive Interaction Consider two sensors deployed in an effort to discriminate a target. The first is a ground vehicle equipped with an X-band radar. The second is an airborne asset, also equipped with an X-band radar. The result of both radars being active in the same place and at the same frequency is that neither can sense the target due to in-band radar interference. Each sensor is effectively jamming the other, resulting in a decreased ability to sense the environment. In this case, the result of deploying multiple sensors was worse than deploying either sensor singly.

These two examples demonstrate the need for coordinated solutions in sensor deployments. In order to maximize the rate of information acquisition, the sensors must cooperate both to take advantage of situations in which constructive interaction is possible, and to mitigate situations in which negative interaction occurs.

Encouraging cooperation between autonomous agents is an area of active research both in the areas of engineering and computer science [40], as well as in social science and economics [20]. Much of the previous work in sensor networks has focused on solutions in which only a single sensor can be active at any time [9, 43, 27], which prohibits the possibility of meanigful cooperative sensing. In certain sensor network applications, the assumption that only one sensor is active can be justified, but many sensor networks have the capability to support the deployment and querying of several sensors simultaneously. This will particularly be true for sensor networks for which the main objective is to approximately solve an inference problem as quickly as possible. Such networks can sometimes significantly reduce time by deploying sensors simultaneously in a coordinated manner.

One of the challenges of optimizing joint sensor deployments is the combinatorial growth of the action space, resulting in a very complex global optimization problem. In [33] this global optimization is recaste as a constrained joint optimization problem. Unfortunately, the solution derived depends on isolating control parameters, a user-defined process that significantly limits network autonomy. Also, the final algorithm lacks scalability and can not be easily distributed among sensor nodes. Attempts at solving the global optimization using evolutionary algorithms and particle swarms [35, 28] are similarly limited in application to networks with centralized information and computation. Other attempts at optimal joint deployments were made in [16, 11].

One possible solution to the global optimization problem is to apply the principles of welfare economics and game theory to sensor networks. The applicability of game theoretic solutions to problems of measurement acquisition, aggregation and routing in sensor networks has been demonstrated in [24, 17, 8]. Game theoretic models are attractive in that they are highly distributable and often require only limited global information, making them well-suited for solving complicated problems of distributed optimization, such as those found in sensor networks.

The coopertive algorithm described in this chapter uses individual agents IARs as utilities for actions in the joint action space. First, in Section 3.2, the principle of maximum IAR is extended to the space of multiple sensor deployments. Then, in Section 3.3, a method of encouraging cooperation through the use of side payments is set forth. The method allows sensors to increase the desirability of actions for other sensors that are advantageous to itself. In Section 3.4 the side payment principle will be demonstrated using a cannonical, two-sensor system with inherent conflict. Finally, in Section 3.5 limitations and possible extensions of the algorithm will be proposed.

3.1.1 Planning Paradigms

As was pointed out in Section 1.2 due to the stochastic nature of the inference problems they attempt to solve, real sensor systems can benefit from dynamic feedback. In Chapter 2 dynamic programming was used to find an optimal solution for networks in which no two sensors were simultaneously deployed. Even with the simplifying non-simultaneous assumption the computational load of complete dynamic programming was prohibitive. Therefore an index rule that maximizes the rate of information acquisition was adopted as an approximate solution that is computationally feasible.

In networks when multiple sensors are simultaneously active the computational problems of dynamic programming are exacerbated. Although simplifying assumptions analagous to the non-simultaneous assumption can be made (see Section 3.2.1), the dimensionality curse makes exact dynamic programming infeasible. However, even though exact solutions can not be formulated, the principle of dynamic feedback can still improve system performance.

In [5] a method for scheduling called Open-Loop Feedback Control is suggested as a suboptimal alternative to exact dynamic programming. In OLFC a control is applied (i.e. actions are chosen) without consideration for future information. This is the Open-Loop portion of the planning method. Then, each time future information becomes available, the control creates a new plan based on all available information (including the information that just became available). This is the Feedback portion of the method. Thus in OLFC, as with exact dynamic programming, actions are dependent on observations and the system is reactive to received information; but the computational burden is significantly reduced from exact dynamic programming because of the simplicity of open-loop planning.

We have adopted a solution method similar to OLFC. At each iteration of the planning algorithm a joint action is chosen for the sensor network. Then, when new information becomes available through sensor measurements a new action is chosen. The mechanism for choosing actions is not, however, the exact solution of an optimization problem (as in OLFC). Instead the principles of game theory are employed. The reasons for this choice are discussed in Section 3.2.2.

To summarize: a type of open-loop feedback control will be used in which each time new sensor measurements become available the sensors in the network choose a new deployment configuration by playing a game. The rules of the game are defined in Section 3.2.2. This game will be replayed every time the (global) posterior density function is updated (i.e. each time new sensor measurements become available). This approach is a compromise between optimality and computability.

3.1.2 A Note on Terminology

In the sequel, the terms sensor, agent, and player will be used somewhat interchangably. I've attempted to use "sensor" when referring to the physical realities of the sensor network, "player" when referring to game theoretic principles relating to the network, and "agent" when referring to general principles of autonomy within the network. These terms all refer to the same physical quantity, which is a single sensor that can operate in one of several modes. Also, the term "configuration" is meant to imply the selected mode of operation for a sensor or group of sensors. It is not limited to the physical placement of the sensors, but can include any variable of operation present in the sensor(s).

3.2 Extending Maximum IAR to Multi-Sensor Deployments

When more than one sensor can be active within a network, the question arises of how to formulate the dynamic program in a meaningful way. From a general standpoint, the dynamic equation from Chapter 2 can be recaste in terms of available assets.

$$\begin{aligned} J_t(\pi_t, D_t) &= \min_{a \in A_t} \left(E[t(a) + J_{t+1}(\pi_{t+1}, D_{t+1})] \right) \\ J_N &= \min_{a \in A_N} E[t(a)] \end{aligned} \tag{3.1}$$

where D_t represents the set of deployable sensors and is updated according to the equation

$$D_{t+1} = \{D_t \backslash a_t\} \cup \{s_{z_{t+1}}\}$$

As in Chapter 2, the state variable π_t is the (global) posterior probability distribution for the inference problem and is updated by equation

$$\pi_{t+1} = f(\pi_t, z, p(z|x))$$

where $f(\pi_t, z, p(z|x))$ is a given update function such as Bayes rule.

The action space in Equation 3.1, A_t, is no longer limited to the various modes of each individual sensor, but is now the set of all **combinations** of deployable sensors, with each sensor possibly configured in any of its modes. One of the difficulties in deploying sensor groups is the high dimensionality of the action space [35].

On of the difficulties in analyzing Equation 3.1 is in understanding the cost-to-go. For singly-deployed sensor networks, the cost-to-go ($E[J_{t+1}(\pi_{t+1})]$) was just the expected time from when the chosen sensor reported its measurement until the inference problem was complete. It was shown in Chapter 2 that, given enough distance from the goal, the sensor and mode with maximum IAR would be the optimal one to deploy or query. However, the analysis depended implicitly on the assumption that sensor measurements were "monolithic" (i.e. that all measurements from a single action were received simultaneously). This meant that all the effects of each action would be known **prior** to the next decision period.

3.2.1 Monolithic Joint Deployments

One solution concept to the problem of multiple, simultaneous sensor deployment is to require that all sensors deployed at a given time must complete their sensing prior to redeployment of any new sensor. This solution preserves the "monolithic" measurement model of the single sensor case. In this scenario, each sensor/modality group could be considered as a single super sensor, and (unsurprisingly) results similar to those found in Chapter 2 can be derived. Specifically, given a group of sensors G and a set of deployment parameters M, define the group IAR to be:

$$\rho(G, M) = E[\frac{\mu(G, M, \pi)}{\max_{i \in G} t_i}]$$

where $\mu(G, M, \pi)$ is the decrease in the uncertainty measure due to deploying group G in mode M with probability function π. Then, given the same assumptions as were made in Section 2.4.2, an optimal action is the one that maximizes $\rho(G, M)$.

However, the "monolithic" assumption may be quite limiting in some sensor networks. For example, imagine the cooperative deployment of two sensors s_1 and s_2 with expected measurement times t_1 and t_2 with $t_1 << t_2$. Under the monolithic assumption, s_1 would have to sit idle while s_2 completed its measurements. Only after s_2 had finished could s_1 be redeployed in a new configuration. Since time minimization is the network goal, having a valuable sensor sitting idle represents a significant problem.

Another limitation of the "monolithic" index rule is that it requires global information. Specifically, the global uncertainty measure decrease due to the several sensors in the group must be calculated as well as the maximum expected deployment time of all the sensors in the group. Such global computations fail to utilize the parallel computational ability of sensor networks. Additionally, knowledge in sensor networks is usually distributed, so computations involving global knowledge require a large amount of inter-sensor communication which may severely impact the network power consumption, as well as time productivity due to communication latency. Another problem is that the action space is the set of all possible joint sensor deployments, resulting in the difficult global optimization problem identified in both [33] and [35].

3.2.2 A Game Theoretic Approach to Maximizing Group IAR

A possible solution to the problem of balancing competing sensor utilities is to employ the principles of game theory. Much of the vocabulary for dynamic programming and game theory is similar, although the two fields have very different lineages, including the concepts of actions, strategies and costs or payoffs. While dynamic programming was developed as a method for optimizing sequential decisions [37, 3], game theory was developed to explain economic phenomena [36]. Dynamic programming has found most of its application in engineering, while game theory has been used in macroeconomics and political science. But, as mentioned in Section 3.1, due to its distributed nature and usefulness in analyzing complicated systems, game theory is finding new relevence in the area of autonomous networks.

Game theory posits that interactions between autonomous agents, called players, can be formulated as a mathematical game. Each player has a set of possible actions and a utility that results from taking each action. A player's utility is also a function of the actions of the other players in the game. For a game with N players, a **joint action**, is an N dimensional vector of individual agent actions. The **joint action space** $A \subset \mathbb{N}^N$, is the set of all possible joint actions. The utility function for player

i, $U_i(\mathbf{a}) : A \mapsto \mathbb{R}$. For the developments in this thesis it is assumed that the joint action space is finite.

In the case of sensor networks, the players represent individual sensors. The actions are the various choices of modalities in which the sensors can be deployed and the utility is the benefit to the sensor when deployed under the specified conditions. A natural choice for utility function, considering the development in Chapter 2, is the sensor's IAR. Notice that while s_i's IAR was previously solely a function of its own mode of operation, now there will be a family of IAR's for each mode, corresponding to each of the modalities of all the other sensors. While this represents a significant calculation, it should be observed that it is only necessary to calculate the modified IARs for those sensors and modes that interact with s_i, since only they will cause a variation to its IAR. This can significantly decrease the amount of necessary computation.

Once the utilities have been computed, there are many possibile solution methods for the derived game. These can be broadly grouped into coordinated and uncoordinated solution methods. Uncoordinated methods involve no revelation of any players preferences to any other player, while coordinated methods involve some meaningful inter-player comparison of preferences prior to the execution of the game.

3.2.3 Uncoordinated Solution Methods

Uncoordinated solution methods are simpler than coordinated methods due to the fact that they do not rely on player i having any knowledge of player j's intended action. The two most popular uncoordinated solution methods are minimax and Nash Equilibrium. Other uncordinated solution methods, including methods based on internal models of other players, are covered in [14].

Minimax In this solution method, player i's action is chosen to maximize its guaranteed payoff, or received utility. If $U_i(\mathbf{a})$ is the utility to player i of joint action defined by

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \tag{3.2}$$

where a_i represents the action taken by player i, then the minimax solution for player i is

$$a_i^* = \arg\max_{a_i}[\min_{a_{j \neq i}}[U_i(\mathbf{a})]] \tag{3.3}$$

The minimax solution method is attractive because it avoids situations where other players could adversely affect player i's payoff, and it provides a lower bound on the actual payoff player i will receive. Also, it only makes use of local knowledge: it is unnecessary for player i to know anything about player j's utilities. In a distributed setting where communication is costly, minimax represents an attractive solution.

One critical point concerning minimax is its inability to encourage meaningful cooperation. Since agents take no consideration of other players actions except to minimize the negative effects of those actions, minimax solutions are generally good at avoiding conflict, but fail to encourage constructive cooperation.

Nash Equilibrium The Nash Equilibrium [25] is perhaps the most popular and well-known result in game theory. It states that for certain game formulations there exists a strategy for each player from which no player has an incentive to deviate. It is a self-enforcing solution, where players can act with confidence that opposing (rational) players have no interest in playing a different strategy.

While very powerful as a solution concept, there are limitations to the effectiveness of the Nash Equilibrium in cooperative games. The traditional example of this is the so-called Prisoner's Dilemma. In this two player game, player's must choose one of two actions: cooperate or defect. If both choose to cooperate they are rewarded. If both choose to defect, they are punished. If one player chooses to cooperate while the other defects, the defecting player receives a reward greater than if it had cooperated, and the cooperating player receives a harsher punishment than if it had defected. In this game the Nash Equilibrium is for both players to defect, *even though there exists a solution with greater utility for both players.* Another example where playing the Nash Equilibrium leads to system suboptimality is described in Section 3.4.

A further obstacle to implementation of Nash Equilibrium in the sensor network context is that it requires full knowledge of all players utility functions. The communication required for each sensor to transmit a multi-dimensional function to every other player may be prohibitive.

3.2.4 Coordinated Solutions

The process of players collaborating in order to make decisions about what actions to play is called bargaining or negotiation. Two main divisions of bargaining can be made: axiomatic bargaining and strategic bargaining.

Axiomatic Bargaining In axiomatic bargaining [29] a solution is sought that is "most fair" for all players. The definition of fairness in this context can be seen as an attempt to balance the sacrifices and gains of each player in an effort to improve the social welfare of the group. Axiomatic bargaining solutions include the Nash Bargaining solution, the Kalai- Smorodinsky solution, the Egalitarian solution and the Utilitarian solution. To find these solutions, players (or a central arbitrator) must have knowledge of the full utility space. As with the Nash Equilibrium, this level of global knowledge is prohibitive in many sensor networks where the knowledge and computation should be distributed across the sensor nodes.

Strategic Bargaining Strategic bargaining, sometimes called non-cooperative bargaining, is an attempt by players to increase their payoffs through collaboration. The players intent in participating in the bargaining is not to achieve a "fair" solution, but to maximize its own payoff through encouraging other players to take actions that will benefit it. The market-based solutions examined in [24, 38] fall into this category of bargaining, with the collaboration taking the form of common knowledge of "prices" of goods or information. Strategic bargaining includes auctions as well as other market based mechanisms for resource allocation.

Strategic bargaining in distributed settings will generally use less communication than axiomatic bargaining, because it is not necessary to communicate the entirety of the utility functions for all players. However, players must still reveal sufficient informormation in order to coordinate their chosen action with that of the other players. Strategic bargaining can be seen as a compromise between the communication requirements of uncoordinated solutions and axiomatic bargaining.

The basic element of strategic bargaining is the contract. In most of the literature contracts are made between "buyers" and "sellers." The contract involves the seller allocating goods to the buyer in exchange for some payment. Defining what goods are being allocated and how payment is made are the central issues of strategic bargaining.

3.3 Increasing Sensor Utility through Side Payments

In the time minimization situation explored in Chapter 2 each agent is trying to maximize its rate of information acquisition. In networks where multiple sensors are simultaneously active, a sensor's IAR may depend on the action of other sensors, either because they are configured in a way that decreases the IAR or because if they were configured differently the sensor's IAR would increase. Either way, the sensor needs some method of persuading the other sensor(s) to reconfigure. Examined from a strategic bargaining perspective, the goods in this situation are agreements by some sensor to reconfigure, meaning to change its intended mode of operation in some way. A sensor that chooses to reconfigure is the "seller" in the bargain. In exchange for agreeing to "sell" its choice of mode to accomodate another sensor, the seller must receive some payment from the "buyer" (i.e. the sensor that wants it to reconfigure). This payment takes the form of a credit that can be used in future negotiations. The payment should be at least as much as the difference in the IAR of the seller under its current configuration and its IAR under the reconfiguration. This exchange of credit will be called a "side payment." Some of the benefits and limitations of side payments in contract formation are discussed in [1].

In developing a method of side payments for a general sensor network, we will proceed in steps. First we will consider a simple two-player, two action game. Then we will extend the analysis to two-players with multiple actions and finally multi-player, multi-action games.

3.3.1 Two-Player, Two-Action Game

Two-player, two action games are often analyzed using the payoff matrix. A general payoff matrix for a cannonical game is shown in Table 3.1. where a_i^j denotes action j

	a_2^1	a_2^2
a_1^1	$[U_1(a_1^1, a_2^1), U_2(a_1^1, a_2^1)]$	$[U_1(a_1^1, a_2^2), U_2(a_1^1, a_2^2)]$
a_1^2	$[U_1(a_1^2, a_2^1), U_2(a_1^2, a_2^1)]$	$[U_1(a_1^2, a_2^2), U_2(a_1^2, a_2^2)]$

Table 3.1: Cannonical payoff matrix

taken by player i and $U_i(\cdot, \cdot)$ represents the utility to player i under some joint action. The purpose of player i making a side payment to player j is to increase player j's utility for taking some action. For instance, if player one's preferred joint action was (a_1^1, a_1^2), it might offer player two a side payment ξ_1 in order to make this option more attractive to player two. The resulting enhanced payoff matrix is

	a_1^2	a_2^2
a_1^1	$[U_1(a_1^1, a_1^2), U_2(a_1^1, a_1^2) + \xi_1]$	$[U_1(a_1^1, a_2^2), U_2(a_1^1, a_2^2)]$
a_2^1	$[U_1(a_2^1, a_1^2), U_2(a_2^1, a_1^2)]$	$[U_1(a_2^1, a_2^2), U_2(a_2^1, a_2^2)]$

By increasing the utility for this joint action, player one increases the likelihood of player two choosing action a_1^2. The source of the "extra" utility player one used to make the side payment will be addressed shortly.

Let $\mathbf{a}_i^*$ be the **ideal joint action** for player i. That is

$$\mathbf{a}_i^* = \arg\max_{\mathbf{a} \in A} [U_i(\mathbf{a})]$$

The principle of the ideal joint action is: if player i were able to choose the individual actions of all N players, what joint action would she choose in order to maximize her utility. The use of ideal points in axiomatic bargaining was suggested in [19]. Consider the game defined by the payoff matrix 3.1 with ideal (joint) actions $\mathbf{a}_1^*$ and $\mathbf{a}_2^*$. If $\mathbf{a}_1^* = \mathbf{a}_2^*$ then the players are in agreement and no further coordintation is necessary. If not, the players must attempt to persuade each other to change actions. This persuasion takes the form of side payments. If player i offers a side payment to player j such that

$$U_j(a_i^*) + \xi_i > U_j(a_j^*)$$

then player j will have sufficient incentive to switch actions and a coordinated strategy results.

If side payments are offered blindly it could result in a situation where both players offer large enough side payments to induce the other player to switch actions. To avoid such situations we adopt the following convention.

Convention 3.1 *When two players are offering simultaneous side payments, only the side payment that maximizes $U_j(a_i^*) + \xi_i - U_j(a_j^*)$ will be valid.*

Furthermore, in order to defeat players that attempt to inflate side payments a second convention will be adopted.

Convention 3.2 *Player i cannot receive utility greater than $U_i(\mathbf{a}_i^*)$*

Under this convention, the maximum side payment paid by player i will be equal to $U_j(a_j^*) - U_j(a_i^*)$. And since no side payment less than $U_j(a_j^*) - U_j(a_i^*)$ is sufficient to induce player j to switch actions, player i will either make a side payment of $U_j(a_j^*) - U_j(a_i^*)$ or 0. (In this discussion, *making* a side payment is different than *offering* a side payment, or making a side payment offer. Offers may be as large as a player's credit allows, but the actual amount levied against a player is limited to the actual utility differential for the other player.) These two conventions increase the level of coordination between the players, resulting in increased communication requirements, but they also provide safeguards on the system to prevent suboptimal action choices. They approximately correspond to the rules for a Vickrey auction [40] for control of the other player's action.

In using side payments for coordination an important issue is the source of the utility that is being offered as a side payment. There must be some method of accruing credit that can be offered as a side payment.

One possibility is to distribute utility that would be gained as a result of successful coordination. In this case, the enhanced payoff matrix looks like

	a_1^2	a_2^2
a_1^1	$[U_1(a_1^1, a_1^2) - \xi_1, U_2(a_1^1, a_1^2) + \xi_1]$	$[U_1(a_1^1, a_2^2), U_2(a_1^1, a_2^2)]$
a_2^1	$[U_1(a_2^1, a_1^2), U_2(a_2^1, a_1^2)]$	$[U_1(a_2^1, a_2^2), U_2(a_2^1, a_2^2)]$

where ξ_1 has been subtracted from player one's utility for the joint action (a_1^1, a_1^2). In [32] an axiomatic method of utility division, called the Shapley value, is derived. So one method would be to divide utility according to the Shapley value among the players. In general, however, this would require a large amount of communication.

In our development utility accrues in a credit account that can be used to make future side payments. When player i receives a side payment, the side payment amount goes into account B_i and can be used by player i to make a side payment offer in a later game. Thus, in the above example, if after receiving payment offer ξ_1 player two decides to take action a_1^2, its account B_2 increases by ξ_1.

This method of side payment generation results in a net conservation of credit within the system. New credit is injected into the system when current credit levels are insufficient to establish coordination. If, for instance, ξ_1 was insufficient to induce player two to choose a_1^2, and if player two were similarly unable to persuade player one to its preferred joint action, then the conflict would be solved randomly. No side payment would be made, but credit would accrue to the account of the player whose joint action was chosen against. In this way the system ensures there will be enough credit to coordinate actions between the players.

Pareto Efficiency of Side Payment method

One of the optimality measures used frequently in game theory is Pareto efficiency. Simply stated, a strategy is Pareto efficient (PE) if there does not exist a strategy such that some player's (expected) utility can be increased without decreasing at least one other player's (expected) utility.

The principle of dominance is related to Pareto efficiency. An action $\mathbf{a}$ strongly dominates action $\tilde{\mathbf{a}}$ if

$$U_i(\mathbf{a}) > U_i(\tilde{\mathbf{a}}),\ \forall\ i$$

Action $\mathbf{a}$ weakly dominates action $\tilde{\mathbf{a}}$ if

$$U_i(\mathbf{a}) \geq U_i(\tilde{\mathbf{a}}),\ \forall\ i$$

and

$$\exists\ i \text{ s.t. } U_i(\mathbf{a}) > U_i(\tilde{\mathbf{a}})$$

We will focus primarily on strong dominance, and will sometimes refer to it simply as dominance.

It can be shown that for a repeated play game with stationary payoff matrix the side payment method described in Section 3.3 corresponds to a mixed strategy in the joint action space. Specifically, for the two- player game it corresponds to the strategy of choosing joint action $\mathbf{a}_i^*$ with probability p_i equal to

$$p_i = \frac{U_i(\mathbf{a}_i^*) - U_i(\mathbf{a}_{\bar{i}}^*)}{U_1(\mathbf{a}_1^*) - U_1(\mathbf{a}_2^*) + U_2(\mathbf{a}_2^*) - U_2(\mathbf{a}_1^*)} \tag{3.4}$$

where

$$\bar{i} = \begin{cases} 1 & \text{if } i = 2 \\ 2 & \text{if } i = 1 \end{cases}$$

This mixed strategy will be called the **negotiated mixed strategy** and will be denoted $\mathbf{a}^*$. Determining whether the negotiated mixed strategy is Pareto efficient provides some insight into the optimality of the side payment method.

Consider the two-player, two action game with ideal actions $\mathbf{a}_1^*$ and $\mathbf{a}_2^*$. Assume $\mathbf{a}_1^* \neq \mathbf{a}_2^*$ since if $\mathbf{a}_1^* = \mathbf{a}_2^*$ no negotiation occurs. The expected per game utility to player i of the side payment method or the equivalent negotiated mixed strategy is

$$U_i(\mathbf{a}^*) = p_i * U_i(\mathbf{a}_i^*) + (1 - p_i) * U_i(\mathbf{a}_{\bar{i}}^*) \tag{3.5}$$

with p_i as defined in Equation 3.4.

Without further assumption the distance from the negotatied mixed strategy to the Pareto frontier can be bounded. Specifically, let $d(\mathbf{a}, \mathbf{a}^*)$ denote the Euclidean distance in utility space

$$d(\mathbf{a}, \mathbf{a}^*) = \sqrt{(U_1(\mathbf{a}) - U_1(\mathbf{a}^*))^2 + (U_2(\mathbf{a}) - U_2(\mathbf{a}^*))^2}$$

Since $\mathbf{a}_i^*$ is the pure strategy that maximizes $U_i(\mathbf{a})$, no action can result in utility greater than $[U_1(\mathbf{a}_1^*), U_2(\mathbf{a}_2^*)]$. Let $\mathbf{a}_I$ be the (possibly) fictional action with utility $[U_1(\mathbf{a}_1^*), U_2(\mathbf{a}_2^*)]$. Define $d1 = (U_1(\mathbf{a}_1^*) - U_1(\mathbf{a}^*)$ and $d2 = (U_1(\mathbf{a}_2^*) - U_1(\mathbf{a}^*)$ Then the minimum distance from $\mathbf{a}^*$ to the Pareto frontier

$$\min_{\mathbf{a} \in A_p} d(\mathbf{a}, \mathbf{a}^*) \leq d(\mathbf{a}_I, \mathbf{a}^*) = \sqrt{d1^2 + d2^2}$$

This principle is shown geometrically in Figure 3-1(a).

If there is no pure strategy solution that strongly dominates the negotiated mixed strategy, then a tighter bound can be shown. Specifically, since no pure strategy dominates the negotiated mixed strategy, for all PE pure strategies $\mathbf{a}_{p_i}$, $U_1(\mathbf{a}_{p_i}) \leq U_1(\mathbf{a}^*)$ or $U_2(\mathbf{a}_{p_i}) \leq U_2(\mathbf{a}^*)$. Thus the maximum posssible PE pure strategy utilities will be $[U_1(\mathbf{a}_1^*), U_2(\mathbf{a}^*)]$ and $[U_1(\mathbf{a}^*), U_2(\mathbf{a}_2^*)]$. These are shown as points A and B in Figure 3-1(b). The minimum distance to the Pareto frontier is

$$\min_{\mathbf{a} \in A_p} d(\mathbf{a}, \mathbf{a}^*) \leq d(\mathbf{a}_I, \mathbf{a}^*) = \frac{d1 * d2}{\sqrt{d1^2 + d2^2}}$$

where $\mathbf{a}_I$ is now the action defined by the projection of the negotiated mixed strategy

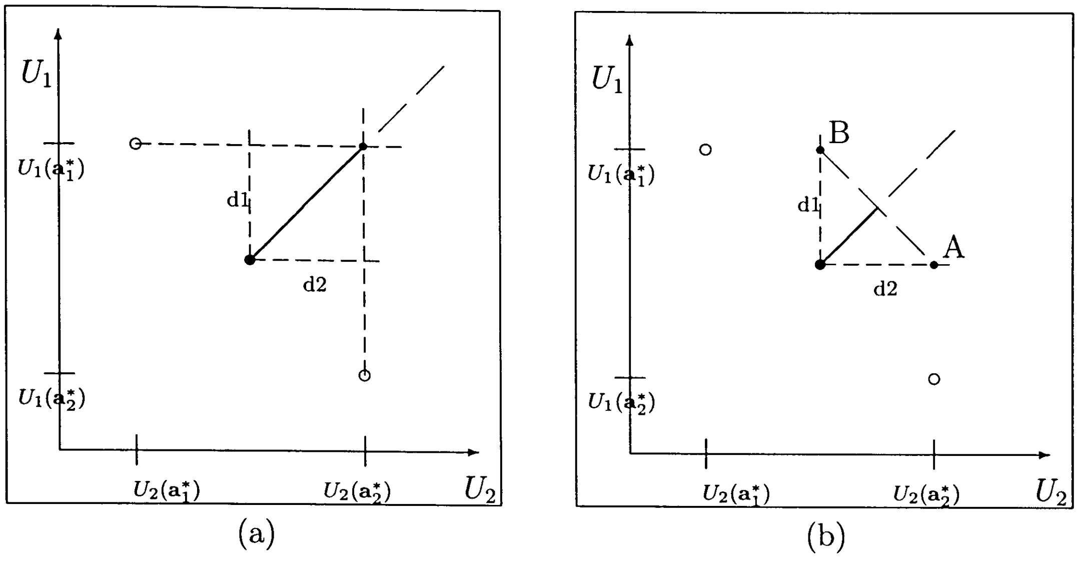

Figure 3-1: (a) Loose bound on Pareto suboptimality of the negotiated solution (b) Tighter bound on Pareto suboptimality when the negotiated solution is undominated

into the line defined by the points $[U_1(\mathbf{a}_1^*), U_2(\mathbf{a}^*)], [U_1(\mathbf{a}^*), U_2(\mathbf{a}_2^*)]$. This is shown in Figure 3-1(b). The bound is improved by at least a factor of 2.

Furthermore, if there exists a pure strategy that lies on or above the line defined by $[U_1(\mathbf{a}_1^*), U_2(\mathbf{a}^*)], [U_1(\mathbf{a}^*), U_2(\mathbf{a}_2^*)]$ then 1) it is a PE strategy and 2) it dominates the negotiated mixed strategy.

The issue of dominance leads to an iterative solution method for the two-player, two-action game.

1. Choose $\mathbf{a}_1^*$ and $\mathbf{a}_2^*$
2. Determine negotiated mixed strategy $\mathbf{a}^*$ using Equation 3.4
3. If undominated, negotiate; otherwise, limit action set to dominating actions and $\rightarrow 1$.

Notice that since there are a total of four possible joint actions, and since $\mathbf{a}_i^*$ can not dominate $\mathbf{a}^*$ for any i, the above algorithm will iterate at most twice. If in step 3 there are dominating pure strategies, these correspond to actions that will be mutually beneficial to both players in the long run. Therefore no side payments should be necessary to induce the players to move from the negotiated mixed strategy to any of the dominating joint actions. Also, it is important to note that just because a strategy dominates the negotiated mixed strategy does *not* imply that it is Pareto optimal. The iterative algorithm will improve the solution suboptimality, but does not guarantee a Pareto optimal solution.

From a distributed computation perspective, an important point is that it is not necessary for player i to know the exact utility of player j for actions other than the ideal actions $\mathbf{a}_1^*$ and $\mathbf{a}_2^*$. This fact limits the amount of global computation and communication necessary. In determining the set of actions that dominate the negotiated mixed strategy, it is sufficient for each player i to broadcast the set of actions A_i such that $\forall \mathbf{a} \in A_i,\ U_i(\mathbf{a}) > U_i(\mathbf{a}^*)$. The intersection of all such broadcast sets is exactly the set of dominating actions. No communication of utilities is necessary.

3.3.2 Two-Player, Multi-Action Games

In light of the algorithm derived at the end of Section 3.3.1, extending the enhanced side payment method to two-player scenarios with multiple actions is elementary. The only difference is that it now may take several iterations of the algorithm before final negotiation occurs. In fact, if the number of actions for player i is N_i then the number of joint actions is $N_1 * N_2$. Each iteration of the algorithm elimiates at least two actions from the feasible set (the two "ideal" actions). Therefore, in the worst case there will be $\frac{N_1 * N_2}{2}$ iterations.

As will be shown in section 3.3.3, the negotiated mixed strategy of each round dominates the negotiated mixed strategy of the previous round (see Equation 3.7). This fact can be used to further limit the amount of communication necessary to determine the set of dominating actions (step 2-3). Because the utility of the negotiated mixed strategy is improving, the set of dominating actions at round k will be a strict subset of the set of dominating actions at round $k-1$. This means that when agent i broadcasts the set of actions with $U_i(\mathbf{a}) > U_i(\mathbf{a}^*)$ at round k, it need only broadcast those that were in the previous round's dominating set.

Generalized Pareto Bound

In the case of the two-player, multi-action game, there may be several rounds of the algorithm before a non-dominated mixed strategy is found. When only one round of negotiation occurs, the bound developed is Section 3.3.1 still applies, but the bound must be modified to be generally applicable. We will delay extending the Pareto bound until after the discussion of multi- player games (Section 3.3.3).

3.3.3 Multi-Player, Multi-Action Games

Generalizing the side payment solution method to games with more than two players, each with an indeterminate number of actions enables application to realistic sensor networks. In this case, at each iteration of the algorithm there are N bargaining agents, each with an ideal joint action $\mathbf{a}_i^*$. Player i offers side payments to all the other players in order to increase the attractiveness of $\mathbf{a}_i^*$. Generalizing Convention 3.1 to the multi-player case,

Convention 3.3 *When N players are offering simultaneous side payments, only the side payment that maximizes* $\sum_{j \neq i} U_j(a_j^*) - U_j(a_i^*) + \xi_i$ *will be valid.*

As in the two-player case, this convention prevents situations where two players are induced to abandon their ideal actions by each other. Convention 3.2 need not be generalized.

The negotiated mixed strategy (i.e. the mixed strategy equivalent in expected utility to the repeated play using side payments) plays action $\mathbf{a}_i^*$ with probability p_i where p_i is given by

$$p_i = \frac{\sum_{i \neq j} U_i(\mathbf{a}_i^*) - U_i(\mathbf{a}_j^*)}{\sum_i \sum_{j \neq i} U_i(\mathbf{a}_i^*) - U_i(\mathbf{a}_j^*)} \tag{3.6}$$

Now the algorithm proceeds as before. We now prove the following useful Lemma.

Lemma 3.1 *At each iteration the Pareto suboptimality of the negotiated solution decreases*

Consider the negotiated mixed strategy at iteration k, $\mathbf{a}^{k*}$. This mixed strategy is a convex combination of the ideal pure strategies of all N players at round k. Let $\mathbf{a}_i^{k*}$

be the ideal action for player i at round k. Since only actions that dominated the $k-1$ round negotiated mixed strategy are considered in round k,

$$\forall\, i,j\ U_i(\mathbf{a}_j^{k*}) > U_i(\mathbf{a}^{(k-1)*})$$

Therefore,

$$U_i(\mathbf{a}^{k*}) = \sum_j p_j * U_i(\mathbf{a}_j^{k*}) > \sum_j p_j * U_i(\mathbf{a}^{(k-1)*}) = U_i(\mathbf{a}^{(k-1)*}) \tag{3.7}$$

Since the expected utility for all players increases with each iteration, the actual distance to the Pareto frontier must be decreasing. This proves Lemma 3.1.

At each round of bargaining, each player chooses an optimal joint action $\mathbf{a}_i^*$. These choices lead to a negotiated mixed strategy $\mathbf{a}^*$. Let the ideal actions at round k be denoted $\mathbf{a}_i^{k*}$ and the negotiated mixed strategy $\mathbf{a}^{k*}$. Because only actions dominating $\mathbf{a}^{k-1}$ are considered at round k,

$$\forall\, i\ \mathbf{a}_i^{k*} \succ \mathbf{a}^{(k-1)*} \tag{3.8}$$

where $\succ$ denotes dominance. Equation 3.8 implies

$$\forall\, i,j\ U_i(\mathbf{a}_j^{k*}) > U_i(\mathbf{a}^{(k-1)*})$$

Furthermore, since the set of actions considered in round k is a strict subset of the actions considered in round $k-1$,

$$\forall\, i\ U_i(\mathbf{a}_i^{k*}) < U_i(\mathbf{a}_i^{(k-1)*})$$

meaning each player's ideal utility is decreasing in the iterations of the algorithm.

Let $\mathbf{U}(\mathbf{a}^{k*})$ be the vector of utilities of the k^{th} round negotiated mixed strategy, $U_i(\mathbf{a}^{k*})$, and define

$$\tilde{\mathbf{U}}_i^k = \begin{bmatrix} U_1(a_1^{k*}) \\ U_2(a_2^{k*}) \\ \vdots \\ U_i(a^{k*}) \\ \vdots \\ U_N(a_N^{k*}) \end{bmatrix} \tag{3.9}$$

That is, $\tilde{\mathbf{U}}_i^k$ is the vector of ideal utilities, except with the i^{th} entry replaced by the utility to player i of the negotiated mixed strategy. Also, define A_k to be the set of actions considered in round k (i.e. the set of all actions such that $\mathbf{U}(\mathbf{a}) > \mathbf{U}(\mathbf{a}^{(k-1)*})$).

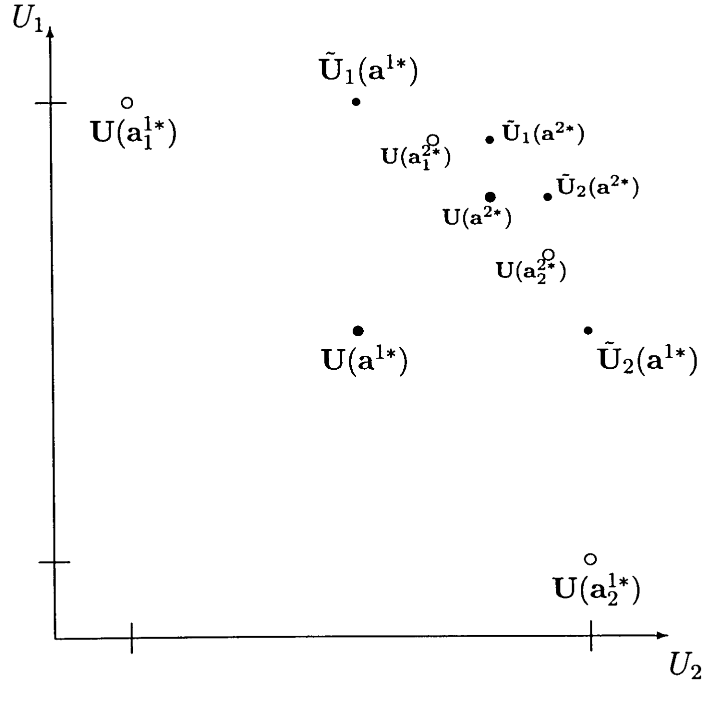

Figure 3-2: Two-player, multi-action utility space

The set $\{\tilde{\mathbf{U}}_i^k\}_i$ defines an $(N-1)$ dimensional hyperplane, which intersects the N-dimensional hypercube defined by the origin and the vector of ideal utilities at points $\tilde{\mathbf{U}}_i^k$.

We would like to show that the Pareto frontier lies within the set defined by $\{\tilde{\mathbf{U}}_i^k\}_{i,k}$ and the positive orthant. We can then bound the pareto suboptimality by the minimum distance between the final negotiated mixed strategy and the hypersurface defined by the boundary of $\{\tilde{\mathbf{U}}_i^k\}_{i,k}$ and the positive orthant.

Consider the set

$$\tilde{\mathcal{U}}_k = \text{conv}(\{\tilde{\mathbf{U}}_i^k\}_i \cup \mathbf{E}_i^k \cup \mathbf{U}(\mathbf{a}^{(k-1)*}))$$

where conv$(\cdot)$ denotes the convex hull and

$$\mathbf{E}_i^k = \begin{bmatrix} U_1(\mathbf{a}^{(k-1)*}) \\ U_2(\mathbf{a}^{(k-1)*}) \\ \vdots \\ U_i(\mathbf{a}_i^{k*}) \\ \vdots \\ U_2(\mathbf{a}^{(k-1)*}) \end{bmatrix} \tag{3.10}$$

with $U_i(\mathbf{a}^{0*}) \triangleq 0$. $\tilde{\mathcal{U}}_k$ is essentially an N-dimensional hypercube with its dominant corner cut off by the hyperplane defined by $\{\tilde{\mathbf{U}}_i^k\}_{i=1}^N$.

Lemma 3.2 *If* $\mathbf{a} \in A_k$ *and does not dominate* $\mathbf{a}^{k*}$ *then* $\mathbf{U}(\mathbf{a}) \in \tilde{\mathcal{U}}_k$

Pf: Assume $\mathbf{U}(\mathbf{a}) \notin \tilde{\mathcal{U}}_k$. Since $\mathbf{a} \in A_k$,

$$U_i(\mathbf{a}) \leq U_i(\mathbf{a}_i^{k*}) \; \forall i$$

by the definition of $\mathbf{a}_i^{k*}$, and

$$\mathbf{U}(\mathbf{a}) \geq \mathbf{U}(\mathbf{a}^{(k-1)*})$$

If $\exists i$ s.t. $U_i(\mathbf{a}) \leq U_i(\mathbf{a}^{k*})$ then

$$\mathbf{U}(\mathbf{a}) \in H(\mathbf{U}(\mathbf{a}^{(k-1)*}), \tilde{\mathbf{U}}_i^k)$$

where $H(x, y)$ is the hypercube with vertices defined by x and y. But

$$H(\mathbf{U}(\mathbf{a}^{(k-1)*}), \tilde{\mathbf{U}}_i^k) \subset \tilde{\mathcal{U}}_k$$

Therefore

$$U_i(\mathbf{a}) > U_i(\mathbf{a}^{k*}) \; \forall i$$

But then $\mathbf{a}$ dominates $\mathbf{a}^{k*}$, which is a contradication.

Using Lemma 3.2 and a backwards induction on the rounds of iteration, we can show that all solutions lie within $\{\tilde{\mathcal{U}}_k\}$. Therefore the Pareto frontier lies within $\{\tilde{\mathcal{U}}_k\}$ which is equivalent to the space defined by the boundary of the convex hull of $\{\tilde{\mathbf{U}}_i^k\}_{i,k}$ and the positive orthant, establishing our result.

Theorem 3.1 *For all* $\mathbf{a} \in A$, $\mathbf{U}(\mathbf{a}) \in \{\tilde{\mathcal{U}}_k\}$.

Pf. Let K denote the final iteration of the algorithm. By definition,

$$\nexists \, \mathbf{a} \in A_K \text{ s.t. } \mathbf{U}(\mathbf{a}) > \mathbf{U}(\mathbf{a}^{K*})$$

Therefore, $\forall \mathbf{a} \in A_K, \mathbf{a} \in \tilde{\mathcal{U}}_K$. Since $\tilde{\mathcal{U}}_K \subset \{\tilde{\mathcal{U}}_k\}$

$$\forall \mathbf{a} \in A_K, \mathbf{a} \in \{\tilde{\mathcal{U}}_k\}$$

For general k, consider $\mathbf{a} \in A_k$. If $\mathbf{a}$ dominates $\mathbf{a}^{k*}$ then $\mathbf{a} \in A_{k+1}$ and by induction $\mathbf{a} \in \{\tilde{\mathcal{U}}_k\}$.

If $\mathbf{a}$ does not dominate $\mathbf{a}^{k*}$ then, by Lemma 3.2 $\mathbf{a} \in \tilde{\mathcal{U}}_k$. Since $\tilde{\mathcal{U}}_k \subset \{\tilde{\mathcal{U}}_k\}$,

$$\forall \mathbf{a} \in A_k, \mathbf{a} \in \{\tilde{\mathcal{U}}_k\}$$

and the induction holds.

We have proved a bound in utility space on the set of all actions. The distance from a chosen action to the Pareto frontier is therefore bounded by the distance from the action to the hypersurface derived in Theorem 3.1. Let $P(\mathbf{U}(\mathbf{a}), F)$ be the orthogonal projection of $\mathbf{U}(\mathbf{a}^*)$ onto face F of the hypersurface. The distance to the Pareto frontier of action $\mathbf{a}^*$ is

$$\min_{\mathbf{a}\in A_p} D(\mathbf{a},\mathbf{a}^*) \leq \min_{F\in\mathbf{F}} D(P(\mathbf{a}^*,F),\mathbf{a}^*)$$

3.4 A Simple Demonstration of Side Payments

Consider a simple network consisting of two non-mobile, active sensors, shown in Figure 3-3. Each sensor can choose to direct its sensing toward one of two locations. Two areas are unique to the individual sensors, while a third is common between the two. If the sensors choose to sense the common area they block each other and neither is able to acquire a good measurement. If the sensors are not directed to the same area they acquire normal measurements. All measurements take exactly one time period, regardless of which area is being sensed, therefore maximizing the rate of information acquisition corresponds to maximizing information gain.

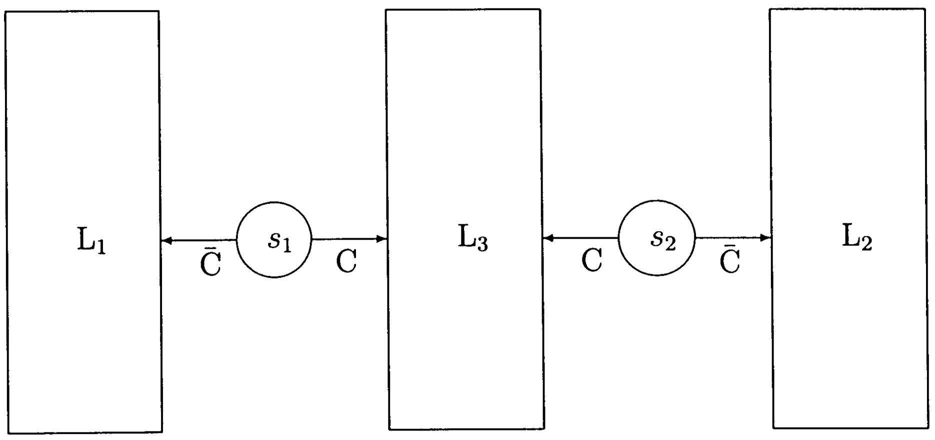

Figure 3-3: Simple cooperative sensor situation

This scenario, although contrived, represents a common situation in sensor networks (and in autonomous networks in general). Often, in order to meet system goals such as quality state estimates or time or energy efficiency, sensors must work both cooperatively and competitively. In the case described above, sensors may be able to increase their IARs if they are able to persuade the other sensor not to sense the common area. By cooperating both sensors can accomplish their goals with greater efficiency.

The payoff matrix for the scenario just described is shown in Table 3.2. Action **C** represents directing sensing to the common area while action $\bar{\mathbf{C}}$ corresponds to directing sensing to the isolated area associated with the sensor. The rows are the actions (or modes of operation) for s_1 and the columns to s_2 and the matrix entries are $(U_1(a), U_2(a))$.

	$\bar{C}$	C
$\bar{C}$	(α_1, β_1)	(α_1, β_2)
C	(α_2, β_1)	$(0, 0)$

Table 3.2: Simple game payoff matrix

The choice of action for this game depends on the relative quantities of $\alpha_1, \alpha_2, \beta_1$ and β_2.

Case 1: $\alpha_1 \geq \alpha_2, \beta_1 \geq \beta_2$ In this case the individually rational solution is $(\bar{C}, \bar{C})$. This represents no conflict and so there is no incentive for either player to make side payments. Each player chooses its maximum IAR. This is somewhat myopic since it may result in a situation where conflict evetually arises. This is a limitation of the single game with side payments method of solution.

Case 2: $\alpha_1 < \alpha_2, \beta_1 \geq \beta_2$ In this case the individually rational solution is $(C, \bar{C})$. There is no conflict in this situation, and both agents choose their maximum IAR.

Case 3: $\alpha_1 \geq \alpha_2, \beta_1 < \beta_2$ In this case the individually rational solution is $(\bar{C}, C)$. Again, no conflict needs to be resolved, and both agents are able to choose their maximum IAR action.

Case 4: $\alpha_1 < \alpha_2, \beta_1 < \beta_2$ This case represents a conflict and hence can demonstrate how the side payment structure from Section 3.3 resolves conflict in sensor networks. For both agents, the maximum utility action is to choose **C**, but if both agents choose that action neither receives any information and they effectively block each other.

There are multiple Nash Equilibria for this case. Two are the pure strategy equilibria: (C,$\bar{\text{C}}$) and ($\bar{\text{C}}$,C). These are what's known as *unstable* equilibria, because any random perturbation of either player's strategy results in convergence to a different equilibrium. This equilibrium, the stable equilibrium, is a mixed strategy, meaning each action is taken with a certain probability. Let $\mathbf{p}$ be player one's strategy with p_1 the probability of taking action $\bar{\mathbf{C}}$ and $p_2 = 1 - p_1$ the probability of taking action $\mathbf{C}$. Similarly let $\mathbf{q}$ be player two's strategy with q_1 the probability of taking action $\bar{\mathbf{C}}$ and $q_2 = 1 - q_1$ the probability of taking action $\mathbf{C}$. It can be shown that the stable Nash equilibrium is $p_1 = \frac{\beta_1}{\beta_2}$ and $q_1 = \frac{\alpha_1}{\alpha_2}$. (If $\alpha_2 = 0$, $\alpha_1 < \alpha 2 \Rightarrow \alpha_1 = \alpha_2 = 0$. Therefore player one has no utility for taking any action and it is a degenerate game. Similarly for $\beta_2 = 0$.)

Notice that if $\alpha_1 = \beta_1 = 0$, meaning neither player has any utility for taking action $\bar{\mathbf{C}}$, the stable Nash Equilibrium is $p_1 = q_1 = 0$ which corresponds to the pure strategy solution (C,C). In this case playing the stable Nash Equilibrium prevents both players from gaining any utility, play becomes deadlocked and *the inference problem is never solved.* Obviously a strategy that plays the Nash Equilibrium will be significantly suboptimal from a system perspective.

Consider the side payment method derived in Section 3.3.1. Since conflict exists in *Case 4* above, under the bargaining method each player will attempt to offer the other a side payment in order to encourage it to switch actions. The revised payoff matrix becomes

	$\bar{\text{C}}$	C
$\bar{\text{C}}$	$(\alpha_1 + \xi_2, \beta_1 + \xi_1)$	$(\alpha_1 + \xi_2, \beta_2)$
C	$(\alpha_2, \beta_1 + \xi_1)$	$(0, 0)$

where ξ_i is the side payment offered by sensor i. If offers are unconstrained, the bargaining could result in a situation in which both sensors would choose to reconfigure (if $\alpha_1 + \xi_2 > \alpha_2$ and $\beta_1 + \xi_1 > \beta_2$). However, generally if offers proceed incrementally, one of the two will exceed the threshold before the other. If not, the two agents have identical preferences and the choice can be made randomly. This payment method can also be seen as a Vickrey auction in which the winning bidder receives the privilege of unimpeded access to the common area. The losing bidder also receives credit equal to the difference between its two utilities. This credit can be applied to resolving future conflicts.

3.4.1 Solution Performance

To demonstrate the effectiveness of the side payment method the cannonical problem described in Section 3.4 was implemented in simulation.

In the simulation the "decrease in uncertainty" for each sensing modality is deterministic but not static. Let n_i^1 be the number of times player i takes action $\bar{\mathbf{C}}$, and let n_i^2 be the number of times player i takes action $\mathbf{C}$ and successfully receives a measurement. To model the fact that each successive measurement has less information content (since it is more likely to include information that was part of a previous measurement) we contract the decrease in uncertainty by θ_d with each measurement. This implies that if the expected decrease in uncertainty prior to the first measurement is $\delta\mu_0$, the expected decrease after n measurements is $\delta\mu_0 * \frac{1-\theta_d^n}{1-\theta_d}$. Additionally, we assume there is some joint information in the measurement area that is common to both sensors, so when one player successfully measures the common area it contracts the possible decrease in uncertainty for the other player by θ_j. Let μ be the current level of uncertainty for each area, with μ_1 corresponding to the area unique to player one, μ_2 the area unique to player two and μ_3 the common area. The system goal can be written as the optimization equation

$$\begin{aligned} a^* &= \arg\min_{a \in A} T(a) \\ &\quad \text{s.t. } \max(\mu(a)) \leq \gamma \end{aligned} \tag{3.11}$$

Using the principle of maximum IAR, and preserving the assumption that every action takes exactly one time step, the individual agents attempt to maximize the information gain at each step. The change in information can be denoted

$$\delta\mu_i^1 = \delta\mu_{0_i}^1 * \theta_d^{n_i^1} \tag{3.12}$$

$$\delta\mu_i^2 = \delta\mu_{0_i}^2 * \theta_d^{n_i^2} * \theta_j^{n_i^2} \tag{3.13}$$

This leads to the payoff matrix for the single stage game of

	1	2
1	$(\delta\mu_1^1, \delta\mu_2^1)$	$(\delta\mu_1^1, \delta\mu_2^2)$
2	$(\delta\mu_1^2, \delta\mu_2^1)$	$(0, 0)$

One reason this scenario was examined was because an optimal solution can be derived without resorting to combinatorial methods, enabling the computation of a suboptimality ratio for each simulated environment. To simulate different environments, the initial uncertainty of each of the three locations was randomly assigned. In all simulations the rates of information contraction were constant ($\theta_d = .9$, $\theta_j = .99$). The results of over 300 monte carlo simulations are shown in Figure 3-4.

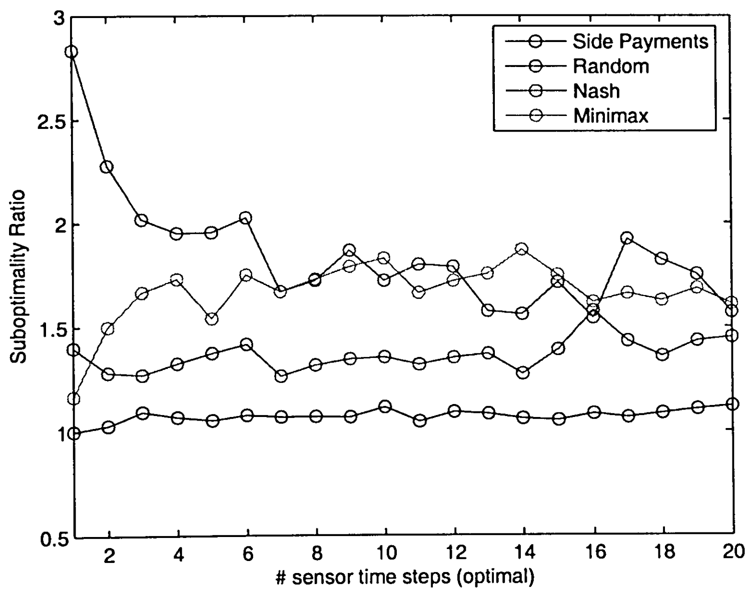

Figure 3-4: Suboptimalities for four algorithms

As is evident from the Figure 3-4, the side payment solution method outperformed the uncoordinated solution methods (random, minimax, and Nash Equilibrium). Additionally, it should be noted that the Nash Equilibrium solution did not result in a feasible solution in 8.9% of simulations due to deadlock, compared with 7.2% for the random algorithm and 2.2% for the minimax algorithm. The side payments solution method found a feasible solution in all of the simulations.

The mean and standard deviation of the suboptimality ratios for the various solution methods are summarized in Table 3.3. Not only was the side payment method closest to the optimal, it also performed the most consistently.

	Side Payments	Random	Nash	Minimax
Mean	1.0733	1.848	1.3363	1.6522
σ_0	0.0839	0.6139	0.2228	0.4971

Table 3.3: Statistical comparison of solution methods

The Nash Equilibrium, minimax and random methods evaluated in Figure 3-4 and Table 3.3 are all uncoordinated methods, so it is perhaps unsurprising that the method of coordination through side payments outperformed them. Further comparison is made in Table 3.4 of the side payment method versus other coordinated methods. In the case of coordinated solution methods, conflicts are resolved through arbitration. The method of arbitration varies with the various solution methods. In NBS the Nash Bargaining Solution (a classical axiomatic bargaining solution) is found. The NBS is a mixed strategy in the dual action space that satisfies specific axioms of "fairness." In the Swap solution method, conflicts are resolved through turn taking. If at the time of the N^{th} conflict player one prevails, player two will receive the advantage in the $(N+1)^{th}$ conflict. This type of solution method takes no account of individual utilities, and so will be of limited use in non- symmetric situations. Again, the algorithms were run on approximately 350 monte carlo simulations with original uncertainties distributed randomly.

The performance differential in this case was much less than with the uncoordinated methods, but the side payment method still outperformed other coordinated

	Side Payments	NBS	Swap
Mean	1.0710	1.0992	1.09462
σ_0	0.0910	0.1210	0.1205

Table 3.4: Statistical comparison of coordinated solution methods

strategies, despite requiring less communication than the Nash Bargaining solution and being more generally applicable than the Swap solution.

3.5 Limitations and Possible Extensions

The above discussion and analysis has focused on the advantages of coordinating sensor measurements through side payments, but there are significant limitations to the method. It does require significant amounts of inter-sensor communication, which may be problematic in networks where power consumption is an issue. The amount of necessary communication can also potentially increase exponentially with the size of the network, since conflicts may arise between any subset of two or more sensors within the network.

The issue of agreement also has not been addressed, specifically as it applies to distributed decision making. It was assumed that sensors did not begin taking measurements until all conflicts have been resolved, all bargains have been made and the network has entered a stable state. Knowing when such a state has been reached is an issue of great practical importance in distributed systems [23].

In extending the above techniques, it may be fruitful to consider strategic players. In the above analysis, sensors took side payments only when the resulting enhanced utility exceeded its currently expected utility. This is what's called a "price taking" technique in the market-based literature. However, in non-stationary networks there may be some benefit to planning when to take on credit based on a network model. The results of [18] may be useful in analyzing such a situation.

The methods developed in this chapter have focused on strategic bargaining to extend the maximum IAR principle derived in Chapter 2 to a setting in which multiple sensors may be active at once. A strategic bargaining soultion was adopted because of its ability to avoid conflict and encourage cooperation without massive communication and computational overhead. The performance of the strategic bargaining method was verified using monte carlo simulation on a cannonical problem of sensor cooperation. In Chapter 4 results will be presented from more varied and realistic sensor situations that will further support the use of side payments for conflict resolution in sensor networks.

Chapter 4

Simulated Implementation and Results

4.1 Simulation Setup

To examine the effectiveness of the MIAR method of choosing sensor parameters developed in Chapter 2 and the multi-agent negotiation protocol described in Chapter 3 several simulations were developed. These experiments and their results are described in this chapter. First the MIAR principle for parameter selection was tested for two different single sensor problems described in Section 4.2, then the application of the negotiation protocol was tested in two sensor network problems described in Section 4.3.

4.2 Single Sensor Simulation

Both simulations described in this section utilize the same basic sensor model and environment, described in the following sections.

4.2.1 Simulated Environment

The simulated environment for these experiments consisted of a 2-D world in which possible target locations were chosen randomly. In a battlefield scenario such "hotspots" can often be determined using pre-deployment intelligence, taking into account such factors as terrain type, road locations, foliage density, river paths and other environmental considerations. Each location may be accompanied by *a priori* probabilities for each target type. Thus targets corresponding to boats are much more likely to be found in rivers than thick forests. This information is reflected in an initial probability vector for each target location.

Figure 4-1 shows a sample problem space for the simulation. In this case, there are ten possible target locations and one target. The target locations are shown as circles while the target itself is shown by a star.

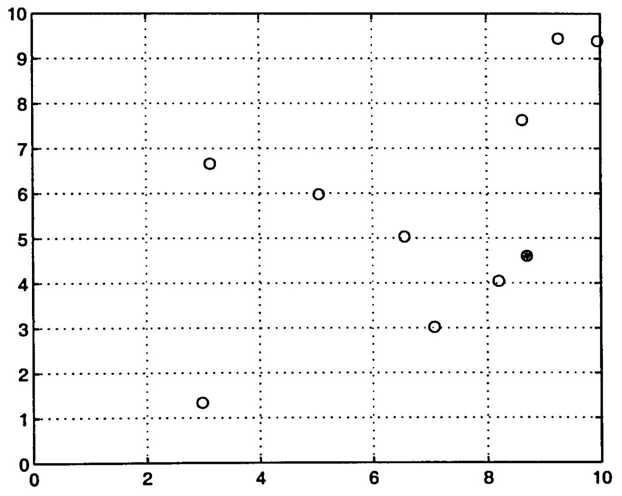

Figure 4-1: Example of a simulated battlefield

4.2.2 Sensor Model

The sensor model consisted of two components: the measurement model and the time model. Deployments of the sensor were assumed to consist of a center point and a sensing radius r. Measurements were received for all locations within distance r

of the center point according to the measurement model described in Section 4.2.2. Also, as described in Section 4.2.2, all measurements for a single deployment become available simultaneously upon completion of the deployment. The total time for the deployment is modeled to depend on the chosen sensing radius r (but not the chosen center point).

Measurement Model

Sensors were simulated to receive power measurements according to the equation

$$z = (\sqrt{\mathrm{SNR}_i(l)} + \eta)^2 \tag{4.1}$$

where for each sensor i, $\mathrm{SNR}_i(l)$ maps the set of target locations to $\mathbb{R}^+$ and η is a zero-mean, unit variance random variable. SNR was chosen to be a function of the survey location l in order to model the effects of target occlusion and obfuscation on sensor management. For instance, most sensors' SNRs are higher for targets in the open than occluded targets (i.e. under camoflage, behind trees, etc.) For stationary sensors, the effect of location on the target SNR may have less to do with occlusion than distance from the sensor. The SNR in this case is assumed to decrease proportionally to the inverse of the square of the distance of the target from the sensor. For each simulation that uses this sensor model the SNRs will be specifically stated.

This measurment model was chosen for two reasons. First, it captures essential sensor differences in a simple, one parameter equation. Second, it is a rough approximation of the actual measurements of several types of active power sensors.

Time Model

The time for a single deployment of the sensor was modeled as the sum of two independent factors: preparation time and sensing time. The preparation time was assumed constant (i.e. identical for each sequential deployment) and independent of the deployment parameters (center point and radius). The sensing time was modeled as a constant times the sensing radius r. For example, if a sensor surveyed locations

(3.2, 5.1) and (7.8, 6.2), the minimum possible deployment radius covering these locations would be $\sqrt{((3.2 - 7.8)^2 + (5.1 - 6.2)^2)/2} = 2.37$. The minimum deployment time to acquire sensor measurements for these two locations would be $a + b * 2.37$. For sensors that require significant amounts of pre- and post-processing, a will be generally be large. Varying b can model different velocities in mobile sensors or the rate of signal propagation for stationary sensors.

Single-mode Simulation

First, the MIAR principle was applied to the problem of choosing the center point and radius for successive deployments of a sensor with a single mode of operation. The perfomance of the MIAR algorithm was compared to two other algorithms: the "random" algorithm, which chose randomly among a set of center points and radii, and the "all in" algorithm which surveyed all locations with each deployment. The problem was to correctly classify each of the randomly selected target locations described in Section 4.2.1 as either being occupied by a target or not. The chosen measure for this problem was the Shannon entropy of each possible target location.

Figure 4-2 shows sample trajectories of the decrease in entropy versus time for the three different assignment algorithms.

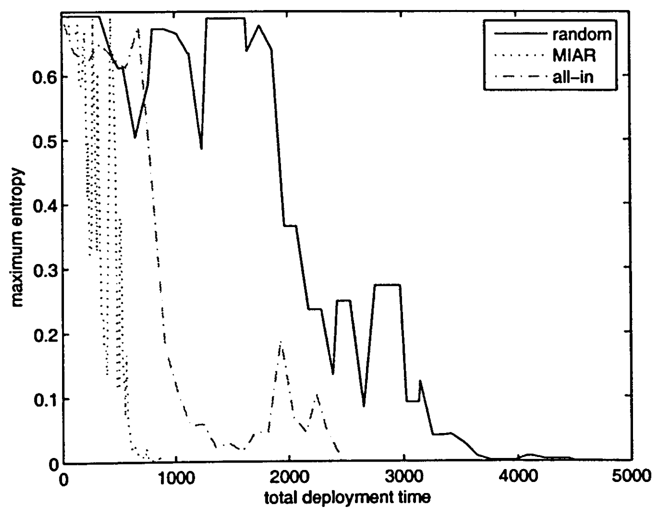

Figure 4-2: Sample entropy trajectories vs. time

Algorithm	Mean Time	σ_o Time	Max Run Time	Min Run Time
Random	11944	3460	20978	6282
All in	4519	1195	8066	2259
MAIR	2519	479	3964	1766

Table 4.1: Statistics for 100 runs of single sensor, single mode simulation

The discrete jumps in Figure 4-2 correspond to the ending of one deployment time and the beginning of the next. Each time new measurements became available they were integrated into the probability mass functions with a corresponding change in entropy at the survey locations.

Statistics for the different algorithms over 100 runs of the simulation are summarized in Table 4.1.

The MIAR algorithm shows dramatic improvement compared to the "all in" algorithm. This demonstrates that, in choosing how to deploy a single, mobile sensor, the maximum IAR principle can improve the time efficiency of solving the inference problem.

Of interest is how the time constants affect the performance of the algorithm. Figure 4-3 shows how the difference between the "all in" and MIAR algorithms depend on the ratio $\frac{a}{b}$. The plots represent the average performance of the "all in" and random algorithms relative to the MIAR algorithm. The error bars delineate two standard deviations.

For values of $\frac{a}{b}$ near zero, total time cost is dominated by sensing time. In this case the difference between MIAR and both alternative algorithms is large. As the ratio increases, the difference decreases. In the limit as $\frac{a}{b} \rightarrow \infty$ the diffence goes to zero. This makes sense, since when $a >> b$, the increase in $E[t]$ due to adding a new location is small compared to the increase in $E[\mu(\pi)]$, so the deployment chosen by the MIAR algorithm will approximately correspond to the "all in" deployment. It is interesting that the difference with the random algorithm also goes to zero. This suggests that as the amount of time dedicated to non-transit activities increases, the utility of any planning decreases. An algorithm that chooses which locations to survey randomly performs as well as either the "all in" algorithm or the MIAR

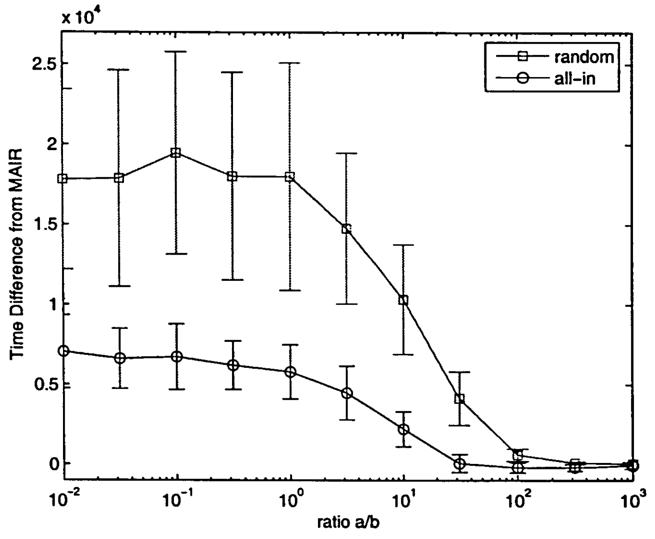

Figure 4-3: Relative Performance Loss as a Function of $\frac{a}{b}$

algorithm. The MIAR appears to be most relevant when the majority of time is spent in surveying and measuring, not in preparation or post-processing.

Multi-mode Simulation

A second single sensor simulation was run to determine the MIAR algorithm's effectiveness for managing modes in a multi-mode sensor. In this case, the single sensor was modeled to have two modes: a wide area mode and a high resolution mode. In the wide area mode, the SNR was lower, but so were the time constants a and b. This represents a fast survey mode that can get measurements quickly, but with limited quality. The high resolution mode, by contrast, takes a long time to get measurements, but the measurements are of a higher quality. Table 4.2 summarizes statistics from this simulation. The MIAR algorithm uses the MIAR principle to choose which mode to use for each successive deployment, as well as which locations to survey. The "high resolution" always chooses the high resolution mode, but uses MIAR to determine the locations to survey. Similarly, the "wide area" always uses the wide area mode, but chooses locations according to the MIAR principle.

Notice that, on average, the MIAR algorithm outperforms each of the modes

Mode	Mean Time	σ_o Time	Min Run Time	Max Run Time
MIAR	5429	775	3957	7870
high resolution	5667	759	4280	7258
wide area	6529	1174	4583	10511

Table 4.2: Statistics for 50 runs of single sensor, muliple mode simulation

individually. This demonstrates that MIAR is effective at balancing the benefits of multiple modes in a single sensor. Also, note that on at least one run the MIAR was outperformed by the high resolution only algorithm. The optimality of MIAR is dependent on several factors which were not necessarily satisfied in this experiment, thus it can only be said to be approximately optimal in this case. The trial-by-trial difference between MIAR and the two single mode algorithms is shown in Figure 4-4.

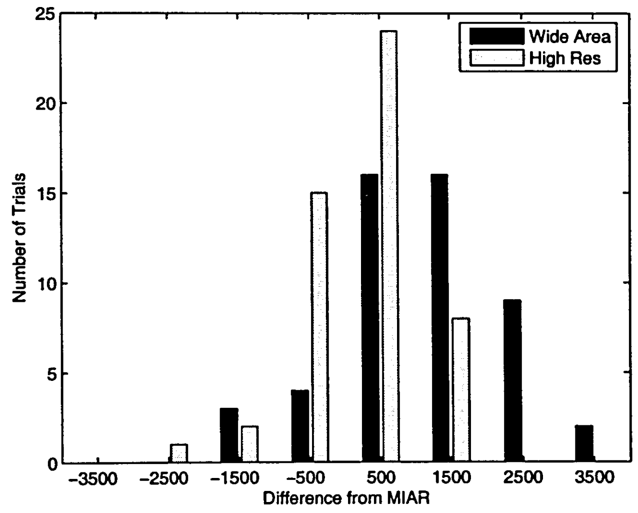

Figure 4-4: Trial-by-trial difference in deployment times

4.3 Multisensor Simulation

Two experiments were run using the negotiation protocol described in Chapter 3. The first was a simulation based on the problem formulation described in Section 3.4 and used two stationary, homogenous sensors. The second was a difficult discrimination problem using a group of heterogenous mobile sensors. The experiments and results

Algorithm	Mean Time	σ_o Time	Max Run Time	Min Run Time
Random	238.79	65.33	439	112
Nash Equilibrium	243.24	230.28	1858	65
Minimax	242.37	71.82	525	111
Exponential Appr.	153.38	41.17	368	61
Nash Bargaining	150.27	38.83	295	57
Multi-agent MIAR	149.57	40.82	367	55

Table 4.3: Statistics for 300 runs of homogeneous, multi-sensor simulation

are described in the following sections.

4.3.1 Homogeneous Multi-Sensor Experiment

The situation described in Section 3.4 was implemented in simulation. Rather than the deterministic, abstract values given previously, the sensors were modeled using Equation 4.1. The task was to determine whether a target existed at each of the three locations. The sensors had an SNR of zero if no target existed and one otherwise. Following the optimization formulation from Chapter 2 the task was declared complete when the entropy of all locations was below a prespecified threshold. Several of the schemes described previously were tested on this environment with results summarized in Table 4.3

The "Exponential Approximation" algorithm in Table 4.3 approximates the optimal algorithm described in Section 3.4. It uses a decaying exponential approximation of the entropy as a function of the number of measurements thus far, and then estimates the remaining number of measurements necessary for each location. It then coordinates the actions for the two sensors based on these estimates.

As pointed out in Section 3.4, the Nash equilibrium solution for this problem can lead to deadlock. Of the 300 simulation runs, 28 (or 8%) resulted in deadlock under the Nash equilibrium assignment. These deadlocks were not included in the results summarized in Table 4.3.

As can be seen from the table, the negotation protocol described in Chapter 3 effectively coordinates the actions of the two sensors. It performs as well as the other two negotiation schemes (Nash Bargaining Solution and Exponential Approximation) with less communication than the NBS and wider applicability than the Exponential Approximation. The performance of the non-coordinated deployment strategies (Random, Minimax, Nash equilibrium) was significantly poorer than that of the coordinated strategies.

An additional experiment was run on the same environment using a variety of target SNRs in order to determine how algorithm effectiveness varied with difficulty of detection (lowering the target SNR increases the difficulty of detecting the target). The mean times for each algorithm on 50 runs each of 16 different SNRs are plotted as a function of increasing SNR in Figure 4-5.

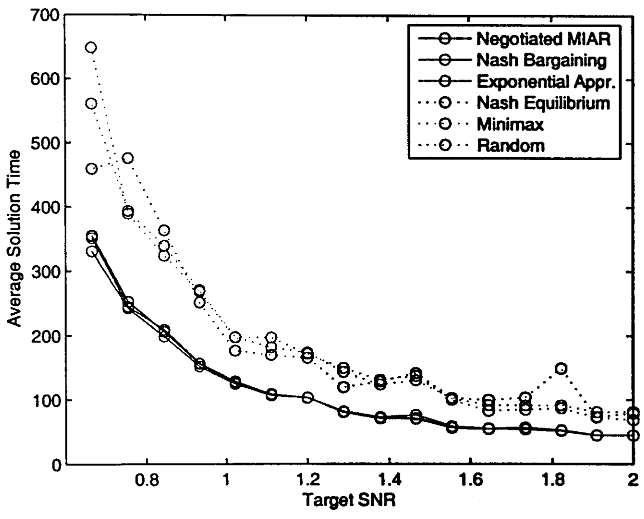

Figure 4-5: Algorithm effectiveness under increasing SNR

The effectiveness of MIAR negotiation appears to be independent of problem difficulty. The most striking feature of the figure is the tight grouping of coordinated and uncoordinated algorithms. The three coordinated schemes were able to significantly improve system performance over the three uncoordinated schemes. In fact, it appears that for this environment, the minimax and Nash equilibrium methods were little better than choosing random deployments, regardless of target SNR. In effect the only real system improvement came through coordination by using a negotiation protocol.

4.3.2 Heterogeneous Multi-Sensor Experiment

To verify the effectiveness of the multi-agent extension of MIAR described in Chapter 3, a discrimination simulation was designed and run. The simulation included five heterogeneous, mobile sensors, each modeled using the sensor model in Equation 4.1. The world was a $5x5$ grid consisting of zero to three point targets of type 1,2, or 3 randomly placed. The goal of the sensor group was to correctly detect and discriminate all of the targets as quickly as possible. The uncertainty measure in this case was the conditional Shannon entropy. The task was declared complete when the entropy of each location was below a predetermined threshold.

The five sensors fell into two broad categories: two detectors and three discriminators. The first detector was constrained to only survey all the possible locations; this could model, for instance, a high-altitude or space-based sensor. The sensing occured quickly (three standard time steps, or a rate of 8.3 locations per time step), but the SNR was low (one if target existed, regardless of taget type, zero otherwise). Additionally, if any of the other sensors were active during the sensing period the SNR decreased to zero, even if a target existed. The second detector could choose any of the nine $3x3$ subgrids to survey. Its SNR was higher (three if target existed, zero otherwise) but took two standard time steps to sense nine locations, or 4.5 locations per time step. Despite being slower, this detector had the advantage that there was no destructive interference from other sensors.

There were three discriminators, one for each target type. For its specific type the SNR was five, while it was one for other two target types and zero if no target existed. The discriminators could only survey one location per standard time step (as opposed to 4.5 for the medium detector and 8.3 for the fast detector). Additionally, if two discriminators were simultaneously deployed to the same location the SNRs of each decreased so all targets had value one, and no target had value zero. If all three discriminators measured the same location at the same time, the SNRs for all sensors and all targets was zero. The statistics for this experiment are summarized in Table 4.4

Algorithm	Mean Time	σ_o Time	Max Run Time	Min Run Time
Random	874.06	146.76	1327	567
Utilitarian	94.74	6.6812	115	81
Multi-agent MIAR	124.52	10.96	165	107

Table 4.4: Statistics for 100 runs of heterogeneous, multi-sensor simulation

The Utilitarian strategy is a standard solution strategy in which the action that maximizes the sum of agent utilities is chosen. Although it outperformed the multi-agent MIAR strategy, it requires the same complete knowledge of the utility space as other negotiation strategies. The relative inefficiency of multi-agent MIAR is the tradeoff for it's significant decrease in communication complexity.

4.4 Results Summary

The four experiments described in this Chapter demonstrate that 1) maximizing the expected rate of information acquisition is a good method for choosing sensor deployment parameters and 2) the iterative negotiation protocol is effective from a system perspective at coordinating the actions of sensors within a group without requiring a large amount of inter-sensor knowledge. Future experiments should develop more involved sensor models and more specific environments in order to determine the applicability of these principles to specific problems of interest such as battlefield awareness in specific environments. Hardware implementation could also help gauge the feasibility of the computational requirements of the algorithms.

Chapter 5

Conclusion

5.1 Summary

The current proliferation of available sensors in military, commercial and industrial environments provides great possibilities and also significant challenges. Sensors acting together in a network can increase the probability of detecting rescue candidates in autonomous search and rescue systems, tracking targets of interest in a battlefield scenario, or discriminating between threatening and non-threatening individuals in a security system. Such applications demonstrate the promise of networked sensing, but the promise comes at a price. As sensors increase in complexity, integrating multiple modes of operation into a single sensor suite, and especially as multiple, heterogeneous sensors are combined into a single network for solving a single inference problem, a method of choosing the best possible modes for individual sensors and combinations of modes for multiple sensors must be devised.

One of the first steps in deriving such a method must be to formalize the network's objectives and constraints, resulting in an optimization problem. Three possible formulations were suggested in Chapter 2: first, those that maximize some inference metric such as negative Shannon entropy or probability of detection while achieving the constraints of limited network resources such as energy, bandwidth or time; second, those that seek to minimize resource consumption given a minimimum constraint on some quality metric; and third, those that combine the quality metric and resource usage in a hybrid formulation. While much of the previous work in using sensor networks to solve inference problems has focused on the first and third of these formulations, this thesis focuses on minimizing resource usage given a minimum quality metric constraint. Specifically, the problem of achieving an inference metric above some threshold in a minimum amount of time is proposed. It was then shown that, under certain simplifying assumptions on the nature of inference metric and the environment, and that no two sensors are simultaneously active, that the optimal configuration is the one that maximizes the information acquisition rate (IAR). This is a problem of immediate applicability in military and commercial security systems, where decisions must be made rapidly and resources deployed to find a solution as soon as possible.

Then, in Chapter 3 the principle of maximizing IAR was generalized to networks in which multiple sensors can be simultaneously active. A game theoretic approach was proposed under which each sensor is a player in the game, with possible actions corresponding to the set of sensor deployment parameters. The problem becomes that of determining an optimal joint action. A coordination protocol was derived using a side payment technique to determine a negotiated mixed strategy based on each players locally determined ideal joint action. The coordination method was then enhanced to include an iterative refinment of the set of possible joint actions. It was proven that for a general class of multi-player, multi-action, non-zero sum games, the pareto suboptimaly (defined by Euclidean distance from the final negotiated mixed strategy to the Pareto frontier) can be bounded.

Finally, the theoretical methods derived in Chapters 2 and 3 were applied in a variety of simulated environments. The results of these simulations were presented in Chapter 4, where it was verified by simulation that the maximum IAR principle, combined with the game theoretic extension to multiple active sensors, is effective at coordinating the sensing efforts of multiple sensors in a variety of situations.

The contributions of this thesis are of theoretical and practical importance. Sensor networks are proliferating, but tools for efficiently using these networks are often lacking. Network usage is often ad hoc and operator driven, necessitating human intervention at a rudimentary level of network operation. The proven ability of the MIAR to quickly and efficiently utilize heterogenous groups of sensors to accomplish an inference task is intended to relieve the burden on sensor network operators through automating the control of sensing. The treatment has been systematic and directed, but is in not complete, as is discussed in Section 5.2.

5.2 Future Work

This thesis leaves several avenues of research open for future work. Finding inference metrics and environments that satisfy the independence assumptions from Chapter 2 is probably not possible. However, it may be possible to find metrics and environments that approximately satisfy the assumptions. Preliminary empirical results for a detection problem with Gaussian noise indicate that the conditional entropy metric adopted for the simulations in Chapter 4 can be approximated by a decaying exponential in the number of sensor measurements given a constant mode of operation. This corresponds to the non-stationary system analyzed in Section 2.4.2.

Another interesting area of further research is in analyzing the suggested solution to the constructed network game. Comparisons, both theoretical and empirical, between the proposed method and other negotiation strategies could be made. Additionally, further analysis could lead to refinements that integrate more complex conditions on the pricing of actions, resulting in different negotiated mixed strategies. Also, it may be interesting to examine how relaxing Conventions 3.1 and 3.2 affect the optimality of the final solution. For instance, if an agent is not constrained to accept an offer that exceeds the difference between its ideal utility and its utility under the offerer's ideal joint action, can the pareto suboptimality still be bounded? This may relate to the $\frac{2}{3}$ bound on strategic pricers derived in [17].

Finally, one of the best methods of furthering the research proposed in this thesis will be in analyzing its effectiveness in real-world sensor networks. This includes finding platforms that can support the computational requirements of the method, defining problem situations that benefit from time minimal analysis, implementing the suggested algorithms and methods, and analyzing the resulting data to verify the effectiveness of the group MIAR principle in directing coordinated sensing in sensor networks. MIT Lincoln Laboratory has dedicated significant resources to this project in its ISDS group. The principles developed here were developed with the intent of future integration into real systems for automatic control of sensor networks. The results of this integration should be significant.

Bibliography

[1] M. Andersson and T. Sandholm. Contract type sequencing for reallocative negotiation. In *20th International Conference on Distributed Computing Systems*, pages 154–160. IEEE, 2000.

[2] A.G. Barto, S.J. Bradtke, and S.P. Singh. Learning to act using real-time dyanmic programming. *Artificial Intelligence*, 72(1):81–138, 1995.

[3] Ronald E. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, NJ, 1957.

[4] Dmitri P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, Massachusetts, second edition, 1999.

[5] Dmitri P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, Belmont, Massachusetts, second edition, 2000.

[6] Dmitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*, volume 3 of *Optimization and Neural Computation.* Athena Scientific, Belmont, Massachusetts, first edition, 1996.

[7] D.A. Castañon. Approximate dynamic programming for sensor management. In *36th Conference on Decision & Control*, pages 1202–1207. IEEE, December 1997.

[8] J.H. Chen, R. Anane, K.M Chao, and N. Godwin. Architecture of an agent-based negotiation mechanism. In *22nd International Confernce on Distributed Computing Systmes Workshops.* IEEE, 2002.

[9] M. Chu, H. Haussecker, and F. Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *Int'l J. of High Performance Computing Applications*, 16(3):90–110, 2002.

[10] Maurice Chu. *A Hierarchical Framework for Constructing Computationally Efficient Algorithms for Distributed Inference Problems.* PhD dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, February 2003.

[11] D.J. Cook, P. Gmytrasiewicz, and L.B. Holder. Decision-theoretic cooperative sensor planning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1013–1023, October 1996.

[12] T. Dean, L.P. Kaelbling, J. Kirman, and A. Nicholson. Planning under time constraints in stochastic domains. *Artificial Intelligence*, 76(1-2):35–74, 1995.

[13] E. Ertin, J.W. Fisher, and L.C. Potter. Maximum mutual information principle for dynamic sensor query problems. In *2nd International Workshop on Information Processing in Sensor Networks.* IEEE, April 2003.

[14] Drew Fudenberg and David K. Levine. *The Theory of Learning in Games.* The MIT Press, Cambridge, MA, May 1998.

[15] John C. Gittens. *Multi-armed Bandit Allocation Indices.* John Wiley & Sons, Chichester, England, first edition, 1989.

[16] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte. Information-theoretic coordinated control of multiple sensor platforms. In *Proceedings of the International Conference of Robotics and Automation*, volume 1, pages 1521–1526. IEEE, September 2003.

[17] R. Johari and J.N. Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research*, to appear 2004.

[18] Ramesh Johari. *Efficiency Loss in Market Mechanisms for Resource Allocation.* PhD dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, May 2004.

[19] E. Kalai and M. Smorodinsky. Other solutions to nash's bargaining problem". *Econometrica*, 43:513–518, 1975.

[20] Ehud Kalai. Solutions to the bargaining problem. In Leonid Hurwicz, David Schmeidler, and Hugo Sonnonschein, editors, *Social Goals and Social Organization: Essays in Memory of Elisha Pazner*, chapter 3, pages 77–107. Cambridge University Press, Cambridge, first edition, 1985.

[21] Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*, volume 21 of *Algorithms and Combinatorics.* Springer-Verlag, Berlin, Germany, second edition, 2002.

[22] J. Liu, D. Petrovic, and F. Zhao. Multi-step information-directed sensor querying in distributed sensor networks. In *International Conference on Acoustics, Speech and Signal Processing*, pages 145–148, April 2003.

[23] Nancy Lynch. *Distributed Algorithms.* Morgan Kaufmann Publishers, Inc., San Francisco, CA, first edition, 1996.

[24] G. Mainland, L. Kang, S Lathaie, D.C. Parkes, and M. Welsh. Using virtual markets to program global behavior in sensor networks. In *Proceedings of the 11th ACM SIGOPS European Workshop.* ACM, September 2004.

[25] John F. Nash. Equilibrium points in n-person games. In *Proceedings of the National Academy of Sciences*, 1950.

[26] R. O'Rourke. Transform and roll out: The usn's approach to change. *Janes Navy International*, April 2004.

[27] Y. Oshman. Optimal sensor selection strategy for discrete-time state estimators. *IEEE Transactions on Aerospace and Electronic Systems*, 30:307–314, April 1994.

[28] D. Penny and M. Williams. A sequential approach to multi-sensor resource management using particle filters. In *Proceedings of the SPIE International Conference on Signal and Data Processing of Small Targets*, pages 598–609. SPIE, 2000.

[29] Hans J.M. Peters. *Axiomatic Bargaining Game Theory.* Kluwer Academic, Dordrecht, Netherlands, first edition, 1992.

[30] R.L. Popp, A.W. Bailey, and J.N. Tsitsiklis. Dynamic airborne sensor resource management for ground moving target tracking and calssification. In *IEEE Aerospace Conference, Vol. 3*, pages 405–415. IEEE, 2000.

[31] M.K. Schneider, G.L Mealy, and F.M. Pait. Closing the loop in sensor fusion systems: Stochastic dynamic programming approaches. In *Proceedings of the American Control Conference*, 2004.

[32] Lloyd S Shapley. A value for n-person games. In H.W. Kuhn and A.W. Tucker, editors, *Contributions to the Theory of Games*, volume 28 of *Annals of Mathematics Studies*, pages 307–317. Princeton University Press, Princeton, NJ, 1952.

[33] D. Sinno and D. Kreithen. A constrained joint optimization approach to dynamic sensor configuration. In *36th Asilomar Conference on Signals, Systems and Computers*, pages 1179–1183. IEEE, November 2002.

[34] Merrill I. Skolnik. *Introduction to Radar Systems.* McGraw-Hill, New York, NY, 1982.

[35] K. Veeramachaneni and L.A. Osadciw. Dynamic sensor management using multi objective particle swarm optimizer. In Belur V. Dasarathy, editor, *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, volume 5434, pages 205–216, Orlando, FL, USA, 2004. SPIE.

[36] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, Princeton, NJ, first edition, 1944.

[37] Abraham Wald. *Sequential Analysis.* Wiley, New York, NY, 1947.

[38] W.E. Walsh, M.P. Wellman, P.R. Wurman, and J.K. MacKie-Mason. Some economics of market-based distributed scheduling. In *18th International Conference on Distributed Computing Systems*, pages 612–621. IEEE, May 1998.

[39] R.B. Washburn, M.K Schneider, and J.J. Fox. Stochastic dynamic programming based approaches to sensor resource management. In *Proceedings of the 5th International Conference on Information Fusion*, pages 608–615, July 2002.

[40] Gerhard Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.* The MIT Press, Cambridge, MA, first edition, 1999.

[41] J.L. Williams, J.W. Fisher III, and A.S. Willsky. An approximate dynamic programming approach to a communication constrained sensor management problem. In *Proceedings of the Eighth International Conference on Information Fusion*, July 2005.

[42] N. Xiong and P. Svensson. Sensor management for information fusion – issues and approaches. In *Information Fusion*, volume 3, pages 163–186, 2002.

[43] F. Zhao, J. Liu, L Guibas, and J. Reich. Collaborative signal and information processing: An information directed approach. *Proceedings of the IEEE*, 91(8):1199–1209, August 2003.