Value of Information Based Distributed Inference and Planning

by

Beipeng Mu

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author .....
Department of Aeronautics and Astronautics
May 24, 2013

Certified by .....
Jonathan P. How
Richard C. Maclaurin Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by .....
Professor Eytan H. Modiano
Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Value of Information Based Distributed Inference and Planning

by

Beipeng Mu

Submitted to the Department of Aeronautics and Astronautics
on May 24, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

In multi-agent applications it is often the case that not all information is equally valuable to the missions, and agents are typically resource limited. Therefore it is important to ensure that the resources are spent on getting and conveying valuable information. This thesis presents efficient distributed sensing and planning algorithms that improve resource planning efficiency by taking into account the obtainable *Value of Information* (VoI) and improve distributed sensing efficiency by ensuring agents only broadcast high value measurements.

The first result focuses on communication efficient distributed sensing algorithms. In particular, agents broadcast their measurements only when the VoI in their measurements exceeds a pre-defined threshold. The VoI threshold is further adaptively adjusted to better balance between the communication cost incurred and the long-term accuracy of the estimation. Theoretical results are presented establishing almost sure convergence of the communication cost and estimation error for distributions in the exponential family. Moreover, an algorithm that automatically forgets old information is also developed to estimate dynamically changing parameters.
Validation through numerical simulations and real datasets show that the new VoI-based algorithms can yield improved parameter estimates than those achieved by previously published hyperparameter consensus algorithms while incurring only a fraction of the communication cost.

The second result focuses on efficient distributed planning algorithms. In particular, in a system with heterogeneous agents, a coupled planning framework is presented that evaluates the sensing/exploration activities by the improvement on mission returns.
Numerical results shows that the coupling between exploration and tasking agents encourages better cooperation between them, thus leading to better performance than decoupled approaches. A hardware testbed is developed to demonstrate the perfor-

mance improvements of the coupled approach in context of distributed planning with uncertain target classifications.

Thesis Supervisor: Jonathan P. How
Title: Richard C. Maclaurin Professor of Aeronautics and Astronautics

Acknowledgments

I would like to give thanks to my thesis supervisor, Professor Jonathan P. How, who gave all the guidance over the past two years. It is his advices and encouragements that directed me overcome all the obstacles and get the results presented in this thesis. I would also like to thank Professor Girish Chowdhary who deeply involved in this work and gave me tremendous help on writing and communications. It is also my ACL colleagues, Alborz Geramifard, Luke Johnson, Dan Levine, N. Kemal Ure and Trevor Campbell that had valuable discussions about my research problems with me and helped me moving forward. Their friendship and companion is very important for my life in MIT.

This research is supported in part by Army Research Office MURI grant number W911NF-11-1-0391.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The increasing availability of compact sensing and processing hardware is fueling a trend in which networks of multiple low-cost unmanned autonomous agents collaborate to perform complex missions [1, 2]. Examples of such missions include aerobiological sampling, persistent surveillance, formation control, distributed resource delivery, and target positioning [3, 4, 5, 6, 1]. While low-cost agents have the potential to yield benefits such as scalability, cost-saving, and resiliency, these agents typically have limited on-board resources, such as computation ability, communication bandwidth, and fuel. Hence, efficient distributed algorithms are needed to ensure that agents can optimally utilize the limited resources to finish their missions, especially in environment with uncertainties.

This thesis focuses on sensing and planning algorithms for multi-agent distributed systems. The tasks in these systems often require the agents to collaboratively sense, estimate, or reach agreement on global parameters/states, such as the states of the environment or shared variables related to task settings and assignments. However, it is often the case that the agents only have limited observability of these parameters, and are constrained in the communication and computation resources. Therefore, efficient algorithms are needed. These algorithms should be able to better use exploration resources to sense the parameters, and use the information gathered to do improve planning. The first key focus of this work is to efficiently use resources to estimate system parameters and the second key focus is to utilize inter-play between

exploration and tasking agents, so information gathered by exploration activities can better benefit the missions of the tasking agents.

1.1 Literature Review

1.1.1 Distributed Sensing

Tasks in cooperative missions often require the agents to collaboratively sense, estimate, or reach agreement on global parameters/states, such as the states of the environment or shared variables related to task settings and assignments [7, 8, 9, 10, 11]. This problem becomes more challenging when there are uncertainties in the environment and there are limited resources. Hence, efficient distributed inference algorithms are needed to ensure that agents can optimally utilize the limited on-board resources while collaboratively estimating global parameters/states.

Many distributed estimation algorithms use the notion of *consensus* to estimate the parameters/states of interest (e.g., [11, 12, 13, 10, 14, 15, 16]). In a typical consensus algorithm, an agent attempts to reach an agreement with its neighbors by performing a sequential update that brings its estimate closer to the states/parameters of (a subset of) all of its neighbors. This process asymptotically converges to the average of all agents' states/parameters under mild assumptions on the connectivity of the communication network formed by these agents. For example, Figure 1-1 depicts a situation in which several networked agents are estimating the distribution of a set of parameters θ. In a consensus framework, all agents would communicate their local parameters to reach consensus on a global estimate. The advantages of a consensus-based approach is that it is fully decentralized, and often requires little computational effort by each agent. However, reaching consensus requires repeated and continuous communication, which can be resource-intensive and it is often the case that not all agents have valuable information to contribute at all times (e.g., the updated states/parameters after new measurements are obtained may not be sufficiently different from the preceding states/parameters, or not all agents are in a good

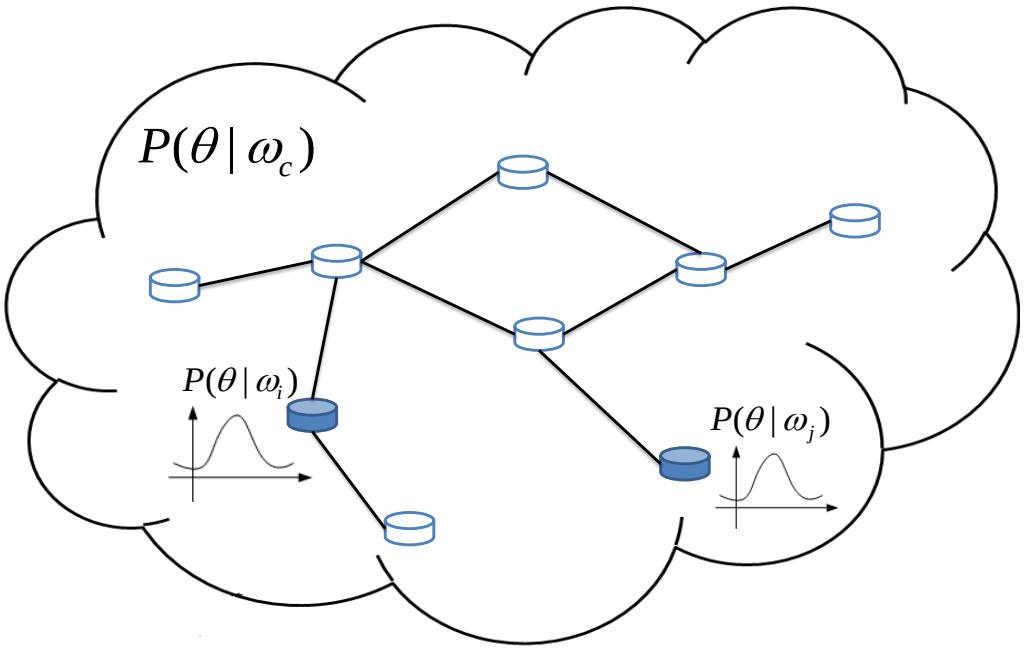


Figure 1-1: Agents communicate local estimates $P(\theta|\omega_i)$ to estimate global parameter of interest θ. A consensus based algorithm requires all agents to communicate their estimates at all times to guarantee asymptotic convergence. However, note that if only two agents (e.g. the dark colored ones) have valuable information, then requiring all the other agents to keep communicating will result in wasted resources.

position to take useful measurements). Thus, requiring all agents to communicate at all times can result in unnecessary communication that clutters the network with marginally useful information.

Revisiting Figure 1-1, we note that if only two agents (dark colored ones) have valuable information, then requiring all the other agents to keep communicating will result in wasted resources. One way to prevent network clutter is to censor (stop) uninformative agents from communicating. However, the consensus framework does not easily allow for dynamic censoring of uninformative agents. This possibly inefficient use of communication resources (and thus energy) could make the implementation of the standard consensus based algorithms difficult in real-world applications.

Another set of algorithms for distributed sensing relies on distributed Bayesian inference using graphical models (e.g., [8, 17, 7, 18]). In graphical model based algorithms, agents build local probability models on the parameters of interest. When

new measurements are observed, agents propagate messages between each other to update their probability models utilizing a priori known information about correlations between each other's probability models. Graphical model based algorithms are only guaranteed to work well on acyclic networks, because in that case there is only one path between any two agents, which guarantees that the messages are not duplicated. For an arbitrary network, one needs to use approximate algorithms (e.g., [19, 20, 21, 22, 23]), or implement additional algorithms to restructure the network into an acyclic network [24], which brings in extra complexity.

Many authors have explored the notion of *censoring* agents/measurements based on some VoI metric to reduce communication cost [25, 26, 27, 28, 29, 30, 31]. Censoring has been mainly explored for centralized estimation frameworks [26, 27]. Cetin et al. have explored censoring in decentralized graphical model-based inference frameworks in the context of a data association problem [25]. In that work, messages are communicated only when the content exceeds a preset VoI threshold. The authors numerically show a significant reduction in communication cost by trading off some estimation accuracy but the paper does not provide theoretical insights on how to choose the VoI threshold.

In contrast, there appears to have been limited work on improving communication efficiency using censoring in the consensus literature. One possible reason for this is that it is not easy to directly apply censoring, such as in [25], to consensus formulations. Censoring agents would result in a dynamic network topology, which could adversely affect the convergence of baseline consensus-based algorithms. In particular, Oliva et al. have stated that adding an agent to a network engaged in consensus would still guarantee convergence to the unbiased global estimate, which is desirable, however, removing an agent from the network introduces a bias [32]. Saligrama et al. introduced a random censoring algorithm aimed at reducing communication cost in consensus based algorithms. In their algorithm, each agent randomly selects a neighbor and passes to it a "transmission permit (token)" [33]. In this way, the communication cost is reduced because not all agents are selected to communicate at

all times. However, that work shows that consensus with only a subset of neighbors communicating takes longer to converge.

1.1.2 Planning with Heterogeneous Agents

Multi-agent missions also require high-level autonomous planning algorithms that coordinate agents with different abilities. These algorithms need to be robust to environmental uncertainties and use information from communication with other agents to avoid conflict and improve performance. Example scenarios include target localization/classification in unknown fields. The problem becomes more challenging when the system has heterogeneous agents with different abilities, in particular, some of the agents are able to finish missions while others can explore the environment to reduce uncertainty.

Agent allocation is traditionally modeled as mathematical programming problems [34, 35, 36]. In particular, robust optimization and stochastic programming techniques are used to deal with parametric uncertainty [37, 38, 39, 40]. Many authors have applied robust optimization techniques to agent planning problems [41, 42, 35]. These frameworks are more robust to environment uncertainties, but typically assume that the uncertainty does not change during the mission. However, it is often the case that in a heterogeneous team, there are some agents that are capable of sensing mission parameters and developing better understanding of the mission environment. The problem of managing agents/sensors to optimally reduce uncertainties in targets/environment is well-studied in sensor management literature [43, 44, 45, 46, 47]. However, in heterogeneous planning problems, the main goal is to maximize the mission return instead of purely reducing uncertainty. Therefore assignments only based on uncertainty reduction can potentially lead to resource waste on highly uncertain but low rewarding missions.

There is relatively limited work that couples uncertainty reduction into distributed multi-agent planning to increase mission returns in a scalable way. Bertuccelli proposed a planning algorithm based on integer programming [35, 48] that establishes a coupling between the exploration and exploitation of missions by a team of het-

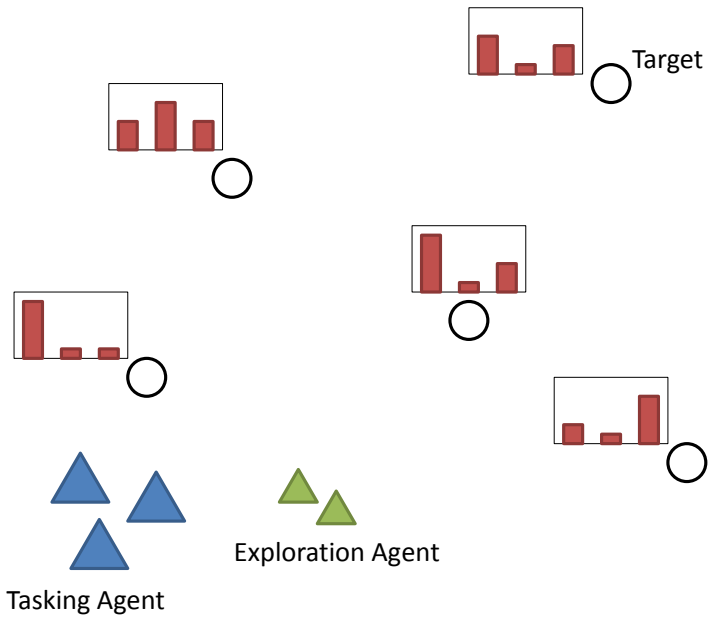


Figure 1-2: Example scenario of heterogeneous agent allocation

erogeneous agents. That work is limited only to uncertainties that have a Gaussian distribution of the potential mission reward. While Gaussian distribution is a good model of uncertainty for measurement noise of continuous states such as positions, orientations and velocities, it cannot well represent discrete uncertainties such as the uncertainty in the category of the mission type. Figure 1-2 depicts a typical situation in which agents (triangles) need to be assigned to targets (circles) that fall into the various category. The uncertainty in this case is the agents' prior guess on which target falls into what category before any measurements are taken, shown as histogram of probability distributions of different categories. In this situation, Gaussian noise would not be the best descriptor of the uncertainty.

1.2 Contributions

This research is motivated by the need to develop more efficient algorithms that are energy efficient and highly scalable for performing distributed estimation and planning than those currently available. The contributions are listed below

- **Distributed Sensing**
  - A Value of Information based Distributed Sensing (VoIDS) algorithm that censors marginally useful information from cluttering the network is developed
  - An Adaptive VoI based Distributed Sensing (A-VoIDS) algorithm that dynamically change VoI threshold to balance communication cost and long-term estimation error is developed
  - A Dynamic VoI based Distributed Sensing (Dynamic-VoIDS) algorithms that can estimate dynamically changing parameters is created.
  - The performance of VoI based sensing algorithms is compare with currently existing distributed sensing algorithms by both simulated as well as real-world data
  - Theoretical bounds on asymptotic performance of VoIDS and A-VoIDS for distributions within exponential family is built
- **Distributed Planning**
  - A Value of Information based Distributed Planning algorithm that couples the value of exploration activities into the mission return of tasking activities in a distributed system with heterogeneous agents is developed
  - Numerical studies with a categorical uncertainty in tasks are conducted to compare performance of coupled and decoupled planning algorithms
  - A hardware testbed is further built to evaluate these planning algorithms

The thesis is organized as follows: Chapter 2 introduces the background on information theory, Bayesian inference and graph theory, which will be used to build the mathematical models of the algorithms. Chapter 3 develops the *Value of Information* based algorithms in the context of distributed sensing problems. Chapter 4 presents the *Value of Information* based algorithms in the context of distributed planning problems. Finally Chapter 5 demonstrates the performance of algorithms

developed in Chapter 3 and Chapter 4 by numerical studies on both simulated as well as real-world data. Preliminary results are presented in papers [49, 50, 51, 52].

Chapter 2

Background

This chapter introduces the required background to formulate the development of Value of Information based sensing and planing algorithms. In particular, it first presents the Bayesian inference, the exponential family, which is used to perform estimation in uncertain environments. Then it presents graph theories, which is used to model communication network of agents.

2.1 Bayesian Inference

2.1.1 Bayes' Law

Bayesian framework is used to model the parameters of interest, because it can describe the uncertainty of parameters with probability distributions and sequentially update the distributions with measurements of the parameters.

Let $\theta \in \mathbb{R}^d$ denote the parameters of interest, $p(\theta)$ denote the prior distribution, and $\mathbf{z} = \{z_1, z_2, \cdots, z_k\}$ denote a set of measurements with the likelihood $p(\mathbf{z}|\theta)$. Bayes' theorem states that the posterior distribution $p(\theta|\mathbf{z})$ is (e.g., [53]):

$$p(\theta|\mathbf{z}) = \frac{p(\mathbf{z}|\theta)p(\theta)}{\int p(\mathbf{z}|\theta)p(\theta)\, d\theta} \tag{2.1}$$

Consistency is one of the basic metrics on performance of estimation problems. It describes when unlimited measurements are used to update the posterior, whether the estimate will converge, and what it will converge to [54].

Definition 2.1 Assume the measurements are independent identically distributed (i.i.d.) drawn from the likelihood function $p(z|\theta_0)$ with parameter θ_o. The posterior distribution $p(\theta|\mathbf{z})$ is said to be consistent at parameter θ_0 if $p(\theta|\mathbf{z})$ converges to Dirac delta function $\delta(\theta_0)$ almost surely (a.s.) when the number of measurements that are used to update the posterior goes to infinity [54].

The Schwartz's consistency theorem outlines a sufficient condition for consistency of Bayesian inference:

Theorem 2.1 (**Schwartz's consistency theorem** [54]). Let $p(x|\theta)$ be a class of probability distributions, $p(\theta)$ be a prior on θ, and $\{z_1, z_2, \cdots\}$ be i.i.d. measurements with likelihood function $p(x|\theta_0)$. Suppose for every neighborhood U of θ_0, every $\theta \in U$ satisfies

$$P\left(\theta : \int p(\theta_0) \log \frac{p(\theta_0)}{p(\theta)} d\theta_0 < \epsilon\right) > 0, \quad \forall \epsilon > 0$$

then the posterior over $\{z_1, z_2, \cdots\}$ is consistent at θ_0. ■

Theorem 2.1 will be used in this paper to test the error of the Bayesian inference framework.

2.1.2 Exponential Family

In general, it is hard or nearly impossible to compute the posterior because the integral $\int p(\mathbf{z}|\theta)p(\theta)\, d\theta$ has no closed-form solutions. However, in the case of exponential family distributions, an easily computable closed-form posterior exists, which gives us an easy way of updating the posterior.

Let $p(\mathbf{x}|\theta)$ denote the probability distribution of random variables $\mathbf{x} \in \mathbb{R}^m$ under some appropriate measure $h(d\mathbf{x})$, given parameters $\theta \in \mathbb{R}^d$. The exponential family

is a set of probability distributions that follow the form [55]:

$$p(\mathbf{x}|\theta) = \exp\left\{\theta^T T(\mathbf{x}) - A(\theta)\right\}, \tag{2.2}$$

where $T(\mathbf{x}) : \mathbb{R}^m \to \mathbb{R}^d$ is the *Sufficient Statistic* or *Potential Function*, and $A(\theta) = \ln \int \exp\left\{\theta^T T(\mathbf{x})\right\} h(d\mathbf{x})$ is the *Log Partition* or *Cumulant Function*. It is proven in [55] that $A(\theta)$ is positive, convex and in class C^∞ within its domain that is well-defined.

The exponential family distributions always have conjugate priors that give closed-form posterior solutions [53]. The conjugate priors are also within the exponential family, with hyperparameters of dimension $d+1$ [55]. Let $\omega \in \mathbb{R}^d$, $\nu \in \mathbb{R}$ denote the hyperparameters and $\Lambda(\omega, \nu)$ denote the conjugate prior's *Log Partition*, then the conjugate prior $p(\theta|\omega, \nu)$ has the following form under appropriate measure $f(d\theta)$:

$$p(\theta|\omega, \nu) = \exp\left\{\theta^T \omega - A(\theta)\nu - \Lambda(\omega, \nu)\right\}. \tag{2.3}$$

For the above exponential family likelihood and conjugate prior, the posterior $p(\theta|\mathbf{z}, \omega, \nu)$ after n measurements $\mathbf{z} = \{z_i\}_1^n$ are observed always has a closed-form solution [56]:

$$\begin{aligned} p(\theta|\mathbf{z}, \omega, \nu) = &\exp\Big\{\theta^T(\omega + \sum T(z_i)) - A(\theta)(\nu + n), \\ &-\Lambda(\omega + \sum T(z_i)), \nu + n)\Big\}. \end{aligned} \tag{2.4}$$

To simplify notation, define augmented vectors $\breve{\omega} = \left[\omega^T, \nu\right]^T$, $\breve{\theta} = \left[\theta^T, -A(\theta)\right]^T$ and $\breve{T}(\mathbf{z}) = \left[\left(\sum T(z_i)\right)^T, n\right]^T$. Then the prior and posterior can be rewritten as:

$$\begin{aligned} p(\theta|\breve{\omega}) &= \exp\left\{\breve{\theta}^T \breve{\omega} - \Lambda(\breve{\omega})\right\} \\ p(\theta|\mathbf{z}, \breve{\omega}) &= \exp\left\{\breve{\theta}^T(\breve{\omega} + \breve{T}(\mathbf{z})) - \Lambda(\breve{\omega} + \breve{T}(\mathbf{z}))\right\} \end{aligned} \tag{2.5}$$

It can be seen that the posterior has the same form as the conjugate prior, only with an additive update in the hyperparameters:

$$\breve{\omega} = \breve{\omega} + \breve{T}(\mathbf{z}). \tag{2.6}$$

The following result can be proven from Theorem 2.1.

Corollary 2.1 If the likelihood is within the exponential family and the prior is conjugate to the likelihood, then the Bayesian inference is consistent.

Corollary 2.1 indicates that when the distribution is within the exponential family, the Bayesian posteriors can get closer estimates of the true parameters by taking more measurements. It will be used to develop theoretical guarantees of the algorithms developed later.

2.2 Graph Theory

Graphs have been traditionally used to describe networked systems. It is also used to provide theoretical guarantees on the performance of these systems.

Let $G\langle v, E\rangle$ represent a graph. Set $v = \{1, ..., N\}$ denotes vertices. Set E denotes edges, $E \subset v \times v$. When $(i, j) \in E$, vertex j is called a neighbor of vertex i. The set of all of i's neighbors is defined as vertex i's neighborhood, denoted by $\mathcal{N}_i$. In real world applications, vertices can represent agents or sensors in a distributed system while edges can represent communication links between them. For example, vertex pair $(i, j) \in E$ if and only if the agents i can communicate with, or otherwise sense, the state of agent j [13].

One efficient way to represent a graph is adjacency matrix. Let $\mathcal{A} = \{a_{ij}\}_{N\times N}$ represent graph $G\langle v, E\rangle$. Element a_{ij} is positive when $(i, j) \in E$, otherwise $a_{ij} = 0$.

$$\mathcal{A} = \{a_{ij}\}_{N\times N}, \qquad a_{ij} \begin{cases} > 0 & \text{if } (i,j) \in E \\ = 0 & \text{if } (i,j) \notin E \end{cases} \tag{2.7}$$

Table 2.1: Information Metrics

Metric	Formula
Kullback-Leibler	$D_{\mathtt{KL}}(p\|\|q) = \int p \log(\frac{p}{q}) dx$
Renyi	$D_\alpha(p\|\|q) = \frac{1}{\alpha-1} \log \int p^\alpha q^{1-\alpha} dx,\ \alpha > 1$
Chernoff	$D_c(p\|\|q) = \log \int p^\alpha q^{1-\alpha} dx$
f-divergence	$D_f(p\|\|q) = \int f(\frac{p}{q}) dq(x)$
Varational	$V(p\|\|q) = \int \|p - q\| dx$
Matusita	$D_M(p\|\|q) = \left[\int \|p^{\frac{1}{r}} - q^{\frac{1}{r}}\|^r dx\right]^{\frac{1}{r}},\ r > 0$

$p(x)$ and $q(x)$ are two probability distributions

Matrix $\mathcal{A}$ is defined as stochastic matrix when $\sum_j a_{ij} = 1, \forall i$. It can be easily proven that a stochastic matrix always has an eigenvalue of 1.

2.3 Value of Information Metric

In estimation and inference problems, the idea of quantifying information dates back to Shannon's information theory [57]. Motivated by Shannon's entropy, Kullback and Leibler introduced the information measure on discrimination between two distributions, now known as the Kullback-Leibler (KL) divergence [58, 59]. Renyi generalized KL divergence by introducing an indexed family of similar divergence measures [60]. Chernoff independently introduced another family of information metric, known as Chernoff distance, which is different from Renyi divergence only by a multiplicative constant [61]. Further generalization beyond Renyi includes f-divergences (or Ali-Silvey divergences, [62]). These as well as some other metrics are listed in Table 2.1.

For planning problems, if the mission for the agents is to gather information to maximize uncertainty reduction, such as that in surveillance, tracking problems, then information metrics mentioned above can also be used to quantify planning performance [63, 64, 65, 66].

2.3.1 KL Divergence and Bayesian Inference

The metrics in Table 2.1 do not have closed-form solutions for general probability distributions. A VoI metric with a closed form solution is desirable, as it would allow VoI to be computed without requiring a costly sampling procedure. If the probability distribution is within the exponential family, Renyi divergence and related metrics have a closed-form solution, thus using Renyi divergence for VoI can help reduce computational cost. Note that KL divergence is Renyi divergence when $\alpha \to 1$. Here we pick KL divergence to be the metric on VoI in our problem. However, other VoI metrics can also be used with the algorithms developed later.

Recall that $p(\mathbf{z}|\theta)$, $p(\theta|\breve{\omega})$ and $p(\theta|\mathbf{z}, \breve{\omega})$ denote the likelihood, the prior distribution and the posterior distribution respectively. If the prior is conjugate to the likelihood as defined in (2.3), Nielsen and Nock show that the KL divergence between the prior and the posterior is [67]:

$$D_{\mathtt{KL}}\left(\ p(\theta|\breve{\omega})\ ||\ p(\theta|\mathbf{z},\breve{\omega}\ )\right) = \Lambda\left(\breve{\omega} + \breve{T}(\mathbf{z})\right) - \Lambda\left(\breve{\omega}\right) - \breve{T}(\mathbf{z})^T \nabla\Lambda\left(\breve{\omega}\right)$$

where ∇ represents gradient. Because $\Lambda(\breve{\omega})$ is in the class C^∞ [55], $\Lambda\left(\breve{\omega} + \breve{T}(\mathbf{z})\right)$ can be expanded in a Taylor series around $\Lambda\left(\breve{\omega}\right)$:

$$\begin{aligned}
&D_{\mathtt{KL}}\left(\ p(\theta|\breve{\omega})\ ||\ p(\theta|\mathbf{z},\breve{\omega})\ \right) \\
&= \Lambda\left(\breve{\omega} + \breve{T}(\mathbf{z})\right) - \Lambda\left(\breve{\omega}\right) - \breve{T}(\mathbf{z})^T \nabla\Lambda\left(\breve{\omega}\right) \\
&= \left\{ \Lambda\left(\breve{\omega}\right) + \breve{T}(\mathbf{z})^T \nabla\Lambda\left(\breve{\omega}\right) + \int_0^{\breve{T}(\mathbf{z})} (\breve{T}(\mathbf{z}) - x)^T \nabla^2\Lambda(\breve{\omega} + x)dx \right\} \\
&\quad - \Lambda\left(\breve{\omega}\right) - \breve{T}(\mathbf{z})^T \nabla\Lambda\left(\breve{\omega}\right) \\
&= \int_0^{\breve{T}(\mathbf{z})} (\breve{T}(\mathbf{z}) - x)^T \nabla^2\Lambda(\breve{\omega} + x)dx \\
&= \frac{1}{2}\breve{T}(\mathbf{z})^T \nabla^2\Lambda(\breve{\omega} + \delta\breve{\omega})\breve{T}(\mathbf{z})
\end{aligned} \tag{2.8}$$

where $\delta\breve{\omega} \in \left[0, \breve{T}(\mathbf{z})\right]$. It is further proven in [55] that $\nabla^2\Lambda(\breve{\omega}) = \mathtt{cov}(\breve{\theta}|\breve{\omega})$, therefore,

$$D_{\mathtt{KL}}\left(p(\breve{\theta}|\breve{\omega})||p(\breve{\theta}|\mathbf{z},\breve{\omega})\right) = \frac{1}{2}\breve{T}(\mathbf{z})^T\mathtt{cov}(\breve{\theta}|\breve{\omega}+\delta\breve{\omega})\breve{T}(\mathbf{z}) \tag{2.9}$$

Lemma 1 Assume the likelihood $p(x|\theta)$ is within the exponential family. Denote the conjugate prior as $p(\theta|\breve{\omega})$, and posterior after taking n measurements $\mathbf{z} = \{z_i\}_1^n$ as $p(\theta|\breve{\omega}+\breve{T}(\mathbf{z}))$. If all measurements are i.i.d. drawn from a distribution with static parameter θ_0, $z_i \sim p(x|\theta_0)$, then $\lim_{n\to\infty}\mathtt{cov}(\breve{\theta}|\breve{\omega}+\breve{T}(\mathbf{z})) \to 0$ a.s.

Proof 1 From corollary 2.1,

$$\lim_{n\to\infty} p(\theta|\breve{\omega}+\breve{T}(\mathbf{z})) \to \delta_{\theta_0} \quad \text{a.s.}$$

Then,

$$\begin{aligned}\lim_{n\to\infty}\mathtt{cov}(\breve{\theta}|\breve{\omega}+\breve{T}(\mathbf{z})) &= \lim_{n\to\infty}\int\left[\breve{\theta}^2 - (\mathbb{E}\breve{\theta})^2\right]p(\theta|\breve{\omega}+\breve{T}(\mathbf{z}))df(\theta).\\ &\to \int\left[\breve{\theta}^2 - (\mathbb{E}\breve{\theta})^2\right]\delta_{\theta_0}df(\theta) \quad \text{a.s.}\end{aligned}$$

since $\breve{\theta}$ is a function of θ and θ_0 is a static parameter,

$$\begin{aligned}&= \breve{\theta}_0^2 - (\mathbb{E}\breve{\theta}_0)^2 \quad \text{a.s.}\\ &= 0 \quad \text{a.s.}\end{aligned}$$

■

2.3.2 Pre-active Planning

For planning problems, when the mission return is not directly measured by uncertainty, entropy and divergence is not long a good candidate to evaluate information. For these cases, there is relative limited work on how to connect information gathering to agents assignments. In Bertuccelli's work [48], the mission goal is to maximize re-

wards by doing tasks at targets, and the potential reward at a target is characterized by a Gaussian random variable C with mean μ and variance σ^2, $C \sim \mathcal{N}(\mu, \sigma^2)$. The planner generates plans that maximizes score $c = \mu - \sigma$. When measurements are taken, the distribution of C becomes $C \sim \mathcal{N}(\bar{\mu}, \bar{\sigma}^2)$ and the score becomes $c = \bar{\mu} - \bar{\sigma}$. The value of these measurements are the increase in mission scores $\bar{\mu} - \bar{\sigma} - (\mu - \sigma)$.

Similarly with Bertuccelli's work [48], information gathering is connected to mission goals by assigning exploration resources based on increase of mission return.

Assume for now discrete time steps and they are index by an integer t. At time t, let $C_{i,t}$ denote the score distribution of target i. If a measurement $z_{i,t}$ of this target is taken, the posterior score distribution, or the score distribution at time instant $t+1$ is the following by Bayes law:

$$p(C_{i,t+1}) = p(C_{i,t}|z_{i,t}) = \frac{p(z_{i,t}|C_{i,t})p(C_{i,t})}{\int p(z_{i,t}|C_{i,t})dp(C_{i,t})} \tag{2.10}$$

Let $f()$ denote a mapping from the probability distribution of mission returns to a utility score. $f()$ should reflect the value of doing missions, therefore higher expectation and lower uncertainty should leads to bigger $f()$. For example, in [48], $f(C) = \mu - \sigma$, where $C \sim \mathcal{N}(\mu, \sigma^2)$. In general, $C_{i,k+1}$ is a function of $z_{i,t}$, therefore $f(C_{i,t+1})$ is also a function of $z_{i,t}$. During the planning stage, $z_{i,t}$ is not available since the measurement is not actually taken yet. A widely used technique in sensor management literature [43, 44, 45, 46, 47] is to consider all possible outcomes of $z_{i,t}$ and compute expected $f()$ on $z_{i,t}$, denoted as $\mathbb{E}f(C_{i,t+1})$:

$$\mathbb{E}f(C_{i,t+1}) = \mathbb{E}\left[f(C_{i,t}|z_{i,t})\right] = \int f(C_{i,t}|z_{i,t})dp(z_{i,t}) \tag{2.11}$$

where $p(z_{i,t}) = \int p(z_{i,t}|C_{i,t})dp(C_{i,t})$. Probability $p(z_{i,t}|C_{i,t})$ represents the measurement model, it gives the likelihood of observing score $z_{i,t}$ when the true score is $C_{i,t}$. For example, if the measurement has Gaussian noise, $p(z_{i,t}|C_{i,t}) \sim \mathcal{N}(C_{i,t}, \sigma^2)$.

If an exploration agent is assigned to take measurements of a target, the potential score to do a mission at this target will change from $f(C_{i,t})$ to $\mathbb{E}f(C_{i,t+1})$. The value of information obtained by assigning an agent to explore this target is then

the increment in the score function: $\mathbb{E}f(C_{i,t+1}) - f(C_{i,t})$. This notion of value of information based pre-active planning can be used to encourage cooperation between heterogeneous agents in a distributed system and improve mission returns.

2.4 Summary

This chapter stated how to collect information and do inference under uncertain environment. Value of Information (VoI) metrics are built to measure the effectiveness of measurement activities. These metrics will be used to guide sensing and tasking activities of agents, which will be articulated in Chapter 3 and Chapter 4

Chapter 3

Value of Information Based Distributed Sensing

This chapter develops the algorithms for distributed sensing problems. In particular, Value of Information based Distributed Sensing (VoIDS) measures the value in new information and selectively share only important messages among the network. However, there exists a trade-off between estimation error and communication cost, therefore Adaptive VoIDS (A-VoIDS) is developed to balance this trade-off by adaptively adjusting the VoI threshold. Furthermore, to deal with dynamically changing parameters, a dynamic VoIDS filter is developed.

Algorithms are tested with simulated data as well as the Intel Data temperature data. Theoretical bounds of VoIDS and A-VoIDS are given to establish almost sure convergence of estimation error and communication cost.

3.1 Distributed Algorithms

First some assumptions used by different distributed inference algorithms are described. Algorithms developed later depend on several or all of these assumptions.

Assumption 1 Graph G is strongly connected. That is, for every vertex pair (i, j), there exists a path from i to j, which can be formed using elements in E.

Algorithm 1 Full Relay

```
initiate global priors p(θ)
for t do
    for each agent i do
        take measurement z_i[t]
        broadcasts z_i[t] to neighbors
        relay received new message z_j[t] to neighbors
        for each broadcast message z_i[t] do
            update the global posterior p(θ|z_i[t]) (2.1)
        end for
    end for
end for
```

Assumption 2 Every agent has a unique identifying label that it can transmit to differentiate its message from others.

Assumption 3 Relaying a message is much faster than obtaining a local measurement, processing it, and then broadcasting it.

Assumption 4 The network topology is known. Or the graph can be uniquely specified.

3.1.1 Full Relay

Based on Assumptions 1–3, a naive method for distributed inference is that every time an agent gets a new measurement, it broadcasts the measurement to all of its neighbors. Furthermore, each agent relays messages for other agents. In this way, all agents have access to all the information from others, essentially allowing every agent to act as the *center* of the network.

Assume that the network is synchronized and the time is indexed by an integer $t \in \mathbb{N}$. Let $m_i[t]$ denote the number of measurements agent i gets at t, and $\mathbf{z}_i[t] = \{z_i^1[t], z_i^2[t], \cdots, z_i^{m_i[t]}[t]\}$ denote the measurements agent i takes at t, the Full Relay algorithm is given by Algorithm 1. It should be noted that this algorithm can be easily extended to asynchronous scenarios.

Cost: The Full Relay algorithm makes a copy of all measurements over each agent. This could lead to big waste in communication resources. Assume that the cost for an agent to broadcast one message to its neighbors is 1 unit. At every time step, each agent needs to broadcast its own message and relay messages for all other agents. The total number of messages every agent sends out at t is N. The step communication cost at each time t (total number of messages sent out by all agents at t) is therefore N^2.

3.1.2 Hyperparameter Consensus

In consensus-based methods, each agent computes an average value between its own estimation and estimations from its neighbors. At each time step, an agent only sends out its local estimate instead of relaying all messages for others. Consensus algorithms are proven to asymptotically converge to global averages (e.g., [68, 11, 12, 13]).

One example of the consensus-based algorithms is Fraser et al.'s Hyperparameter Consensus (HPC) [15]. HPC works on parameter distributions in the exponential family, and performs consensus on hyperparameters. In addition to Assumptions 1 and 2, this algorithm further assumes that the network topology is known. This assumption can be restrictive in some scenarios, but can be relaxed by using topology identification algorithms (e.g., [69, 70, 71]).

Notations t and $\mathbf{z}_i[t]$ are defined the same as in Section 3.1.1. Let $\breve{\omega}_i[t]$ denote the augmented local hyperparameters of agent i at t. Further let $\beta = \{\beta\}_1^N$ denote the eigenvector of eigenvalue 1 of the corresponding adjacency matrix of the network graph; the algorithm is provided in Algorithm 2. Ref. [15] proves that the HPC posterior will asymptotically converge to the centralized Bayesian posterior.

Cost: Noting that at each time step, each agent sends out only one message containing an update of its local hyperparameters, the step communication cost of all agents at time t is N.

Algorithm 2 Hyperparameter Consensus

initiate hyperparameters $\breve{\omega}[0]$
for t **do**
  for each agent i **do**
    take measurement $\mathbf{z}_i[t]$
    compute $\breve{T}(\mathbf{z}_i[t])$
    compute local hyperparameters
      $\breve{\omega}_i[t] = \breve{\omega}_i[t] + \frac{\breve{T}(\mathbf{z}_i[t])}{\beta_i}$
    run consensus algorithm
      $\breve{\omega}_i[t+1] = \breve{\omega}_i[t] + \epsilon \sum_{j \in \mathcal{N}_i} (\breve{\omega}_j[t] - \breve{\omega}_i[t])$
  end for
end for

3.1.3 Random Broadcast

In order to avoid network-wide communication at all times, a random censoring procedure can be used. At every time step, each agent randomly becomes active and sends messages to others. The idea is similar to that in [33] in which each agent randomly select a neighbor to pass a communication token to.

After recording a measurement $\mathbf{z}_i[t]$, instead of broadcasting it immediately, agent i stores it in a local buffer. Define $S_i[t]$ as the sum of the *Sufficient Statistic* of buffered measurements, $n_i[t]$ as the number of buffered measurements of agent i and $\breve{S}_i[t] = \left[S_i^T[t], n_i[t]\right]^T$. Agent i sends out a message containing $\breve{S}_i[t]$ only when a locally generated random number between $[0, 1]$ exceeds a predefined threshold ε. The algorithm is described in Algorithm 3.

Cost: Noting that at each step the probability for an agent to send a message is ε, on average there will be εN agents broadcasting messages. Each message will be relayed by all the other agents, therefore on average the step communication cost would be εN^2. As all agents have a chance to send out their messages, the estimation error is continuously decreasing. By choosing smaller ε, the communication cost would be reduced. However, the convergence rate could also be reduced as agents communicate less frequently [33].

Algorithm 3 Random Broadcast

initiate hyperparameters $\breve{\omega}[0]$
for t **do**
    $\breve{\omega}[t] = \breve{\omega}[t-1]$
    for each agent i **do**
        take measurement and update local buffer
        $\breve{S}_i[t] = \breve{S}_i[t-1] + \breve{T}(\mathbf{z}_i[t])$
        if random number bigger than threshold ε **then**
            broadcasts $\breve{S}_i[t]$
            reset local buffer: $\breve{S}_i[t] = 0$
        end if
        relay received new message $\breve{S}_j[t]$ to neighbors
    end for
    for each broadcast message $\breve{S}_j[t]$ **do**
        update the global posterior
        $\breve{\omega}[t] = \breve{\omega}[t] + \breve{S}_j[t]$
    end for
end for

3.2 Single Agent

Here a network with only a single agent is used to show how VoI can be applied to improve the efficiency of distributed sensing. The VoI based Decentralized Sensing (VoIDS) algorithm for multiple agent network will be given in the next section.

At time t, the hyperparameter of the conjugate prior is:

$$\breve{\omega}[t-1] = \breve{\omega}[0] + \breve{T}(\mathbf{z}[1:t-1]) \tag{3.1}$$

Assume the agent takes $n[t]$ measurements $\mathbf{z}[t]$, $|\mathbf{z}[t]| = n[t]$. The following theorem formalizes the intuitive notion that as an agent takes more measurements, its estimate of the parameters improves, while the VoI in the new measurements $\mathbf{z}[t]$ decreases to zero.

Theorem 3.1 Consider a single agent that takes a finite measurement at every time instance t and all the measurements are i.i.d. drawn from an exponential distribution with static parameters θ_0. At time t, define $V[t]$ as VoI of new measurement

$\mathbf{z}[t]$, i.e, the KL divergence between the conjugate prior and posterior at t, then $\lim_{|\mathbf{z}[1:t-1]|\to\infty} V[t] \to 0$ a.s. ■

Proof 2 From (2.9),

$$\begin{aligned} V[t] =& \breve{T}(\mathbf{z}[t])^T \mathtt{cov}(\breve{\theta}|\breve{\omega}[t] + \delta\breve{\omega})\breve{T}(\mathbf{z}[t]) \\ =& \breve{T}(\mathbf{z}[t])^T \mathtt{cov}(\breve{\theta}|\breve{\omega}[0] + \delta\breve{\omega} + \breve{T}(\mathbf{z}[1:t-1]))\breve{T}(\mathbf{z}[t]) \\ \delta\breve{\omega} \in& \left[0, \breve{T}(\mathbf{z}[t])\right] \end{aligned}$$

Given finite measurements $\mathbf{z}[t]$, *Sufficient Statistic* $T(\mathbf{z}[t])$ is finite, so vector $\breve{T}(\mathbf{z}[t]) = \left[T(\mathbf{z}[t])^T, 1\right]^T$ is also finite. Furthermore, from Lemma 1, $\lim_{|\mathbf{z}[1:t-1]|\to\infty} \mathtt{cov}(\breve{\theta}|\breve{\omega}[0] + \delta\breve{\omega} + \breve{T}(\mathbf{z}[1:t-1])) \to 0$ a.s. Hence

$$\lim_{|\mathbf{z}[1:t-1]|\to\infty} \breve{T}(\mathbf{z}[t])^T \mathtt{cov}(\breve{\theta}|\breve{\omega}[t] + \delta\breve{\omega})\breve{T}(\mathbf{z}[t]) \to 0 \text{ a.s.}$$

that is $\lim_{|\mathbf{z}[1:t-1]|\to\infty} V[t] \to 0$ a.s. ■

Now consider the case in which the agent does not update hyperparameters immediately after taking a new measurement, but instead stores the measurement in a local buffer and calculates the VoI first. The posterior is updated only when the VoI of the buffered measurements exceeds a threshold V^*. Denote $n[t]$ as the number of buffered measurements at t, $S[t]$ as the sum of the *Sufficient Statistic* of buffered measurements, and $\breve{S}[t] = \left[S[t]^T, n[t]\right]^T$. Define t_k as the k^{th} time the agent updates the posterior. This process is described in Algorithm 4.

The following result guarantees that if an agent uses Algorithm 4 for inference, then the frequency of the posterior updates will decrease with time, because the VoI of new measurements will decrease with time due to Theorem 3.1.

Theorem 3.2 Consider the case where a single agent takes one measurement $z[t]$ at every time instance t and does inference according to Algorithm 4. Assume all the measurements are i.i.d. drawn from a static distribution with parameters θ_0,

Algorithm 4 VoI based Sensing for a Single Agent

initiate hyperparameters $\breve{\omega}[0]$
for t **do**
    $\breve{\omega}[t] = \breve{\omega}[t-1]$
    take measurement and update buffer:
        $\breve{S}[t] = \breve{S}[t-1] + \breve{T}(z[t])$
    calculate VoI:
        $V[t] = D_{\mathrm{KL}}\left(p(\theta|\breve{\omega}[t]) \;||\; p(\theta)|\breve{\omega}[t] + \breve{S}[t])\right)$
    if $V[t] > V^*$ **then**
        threshold reached, update posterior
            $\breve{\omega}[t] = \breve{\omega}[t-1] + \breve{S}[t]$
        reset buffer
            $\breve{S}[t] = 0$
    end if
end for

$z[t] \sim p(z|\theta_0)$. Let t_k be the k^{th} time the agent updates the hyperparameters, and $n[t]$ be the number of measurements buffered at t, then $\lim_{t\to\infty} n[t_k] \to \infty$ a.s.

Proof 3 At time instant t, assume $t \in (t_{k-1}, t_k]$, where t_{k-1} represents the last time the agent updated the posterior, and t_k represents the next time the agent will update the posterior.

Furthermore, $n[t_k]$ represents the number of measurements in the buffer at time t_k, i.e. the measurements taken between t_k and t_{k-1}. Since the agent only takes one measurement at every time step, $t_k = n[t_k] + t_{k-1}$. Therefore, $\lim_{t\to\infty}(n[t_k] + t_{k-1}) = \lim_{t\to\infty} t_k \geq \lim_{t\to\infty} t \to \infty$. We have either $\lim_{t\to\infty} n[t_k] \to \infty$ and/or $\lim_{t\to\infty} t_{k-1} \to \infty$.

(i) In the first case, $\lim_{t\to\infty} n[t_k] \to \infty$, the theorem holds.

(ii) Consider for the sake of contradiction that $n[t_k]$ is bounded, that is $\lim_{t\to\infty} n[t_k] \leq C < \infty$. In this case, it follows that $\lim_{t\to\infty} t_{k-1} \to \infty$. In other words, this means that at time t_k, the number of buffered measurements ($n[t_k]$) is bounded, but the number of measurements the agent has used to update the parameters at the previous step (t_{k-1}) goes to infinity. Since t_{k-1} is unbounded, it follows from Theorem 3.1, $\lim_{|\mathbf{z}[1:t_{k-1}]|\to\infty} V[t_k] \to 0$ a.s., which means $\mathbb{P}(V[t_k] > V^*) \to 0$.

Therefore, there does not exist a finite time t_k such that $V > V^*$, hence $n[t_k]$ cannot be bounded, this is a contradiction.

Hence, it must follow that $\lim_{t\to\infty} n[t_k] \to \infty$ a.s. ■

3.3 VoI based Distributed Sensing (VoIDS)

In this section we develop the VoI based Distributed Sensing (VoIDS) algorithm for a network of multiple sensing agents.

In VoIDS, agents start with the same global prior. This can be accomplished by either externally setting a prior to all agents or through communication between the agents to agree on a global prior, as is done in most distributed sensing algorithms without censoring. Similar to the single agent case, upon obtaining a new measurement, agent i records it into its local buffer instead of immediately broadcasting it to others. Denote $n_i[t]$ and $S_i[t]$ as number and sum of *Sufficient Statistic* of buffered measurements for agent i at time t, and let $\breve{S}_i[t] = \left[S_i[t]^T n_i[t]\right]^T$. Denote $V_i[t]$ as the VoI of agent i's buffered measurements at t.

The algorithm proceeds as follows. If $V_i[t]$ exceeds a predefined threshold V^*, agent i labels itself as informative, otherwise it labels itself as uninformative. All informative agents broadcast a message containing $\breve{S}_i[t]$ to their neighbors, then clear their local buffers and reset $\breve{S}_i[t]$ to zero. Uninformative agents censor themselves from broadcasting their own measurements. All agents relay every message they receive from an informative agent or a relaying agent. Since each agent has a unique identifying label, it can be ensured that messages are not duplicated during relay. By Assumption 1, 2, and 3, all agents are guaranteed to get updates of all informative agents. Then they update their estimates of the global posterior by adding relayed updates to their hyperparameters. The process is described in an algorithmic form in Algorithm 5.

The next theorem shows that the interval between two updates for any agents will go to infinity a.s., which means the average communication cost of each step will approach zero a.s. when using Algorithm 5.

Algorithm 5 VoI based Distributed Sensing (VoIDS)

initiate hyperparameters $\breve{\omega}[0]$
for t **do**
  $\breve{\omega}[t] = \breve{\omega}[t-1]$
  for each agent i **do**
    take measurement and update local buffer
    $\breve{S}_i[t] = \breve{S}_i[t-1] + \breve{T}(z_i[t])$
    calculate VoI of current buffer
    $V_i[t] = D_{\mathtt{KL}}\left(p(\theta|\breve{\omega}[t]) \;||\; p(\theta|\breve{\omega}[t] + \breve{S}_i[t])\right)$
    if $V_i[t] > V^*$ **then**
      broadcasts $\breve{S}_i[t]$
      reset local buffer: $\breve{S}_i[t] = 0$
    end if
    relay received new message $\breve{S}_j[t]$ to neighbors
    for each broadcast message $\breve{S}_j[t]$ **do**
      update the global posterior
        $\breve{\omega}[t] = \breve{\omega}[t] + \breve{S}_j[t]$
    end for
  end for
end for

Theorem 3.3 Consider a network of N agents performing distributed inference with Algorithm 5. Assume the measurements are i.i.d. drawn from a distribution with static parameters θ_0. Denote t_k^i as the k^{th} time agent i sends out a message to update the global hyperparameters, then for any agent i, the number of measurements needed to exceed a predefined VoI threshold V^* will go to infinity, that is $\lim_{t\to\infty} n_i[t_k^i] \to \infty$. ■

Proof 4 First assume the case where agent i does not receive any messages from other agents after t_{k-1}^i. Define the time it sends out next message as $\tilde{t}_k^i = t_{k-1}^i + n_i[\tilde{t}_k^i]$. From Theorem 3.2, $\lim_{t\to\infty} n_i[\tilde{t}_k^i] \to \infty$, so Theorem 3.3 holds.

On the other hand, if agent i receives one or more messages from other agents between t_{k-1}^i and $\tilde{t}_k^i$, the global hyperparameters are updated between t_{k-1}^i and $\tilde{t}_k^i$, thus agent i would have used more measurements to update the hyperparameters by time $\tilde{t}_{k-1}^i$. This would only make $\mathtt{cov}(\breve{\theta}|\breve{\omega}[t])$ converge to 0 faster than in the first case due to Lemma 1. Hence in order to reach the same VoI threshold V^*, agent i

needs to take more measurements. Denote t_k^i to be the time agent i sends out the next message, in this case, $n_i[t_k^i] \geq n_i[\tilde{t}_k^i]$. Since $\lim_{t\to\infty} n_i[\tilde{t}_k^i] \to \infty$, $\lim_{t\to\infty} n_i[t_k^i] \to \infty$.

Hence, in both cases, $\lim_{t\to\infty} n_i[t_k^i] \to \infty$. ■

At time t, let I denote the set of informative agents and $\bar{I}$ the uninformative agents. Define the estimation error $e[t]$ as KL divergence between global posterior and the centralized Bayesian posterior:

$$e[t] = D_{\text{KL}}\left(p\left(\theta|\breve{\omega}[t] + \sum_{j\in I}\breve{S}_j[t]\right) \,||\, p\left(\theta|\breve{\omega}[t] + \sum_{i=1}^{N}\breve{S}_i[t]\right)\right) \tag{3.2}$$

The following theorem shows that the expectation of this error is bounded when using VoIDS (Algorithm 5).

Theorem 3.4 Consider a network of N agents that performs inference with Algorithm 5. At time instance t, if the error $e[t]$ is defined by (3.2), then $\mathbb{E}(e[t]) \leq N^2V^*$. ■

Proof 5 For a single measurement z, denote $\mathbb{E}\breve{T} = \mathbb{E}\breve{T}(z)$ as the expected *Sufficient Statistic*. Then $\mathbb{E}\breve{S}_i[t] = n_i[t]\mathbb{E}\breve{T}$. From (2.8), (2.9) and (3.2), take expectation of $V_i[t]$ and $e[t]$ in terms of $\breve{S}_i[t]$:

$$\mathbb{E}(V_i[t]) = \int_0^{n_i[t]\mathbb{E}\breve{T}} (n_i[t]\mathbb{E}\breve{T} - x)^T \mathtt{cov}\left(\breve{\theta}|\breve{\omega}[t] + x\right) dx$$

change integration variable from x to $y = \mathbb{E}\breve{T} - \frac{x}{n_i[t]}$

$$= (n_i[t])^2 \int_0^{\mathbb{E}\breve{T}} y^T \mathtt{cov}\left(\breve{\theta}|\breve{\omega}[t] + n_i[t]\mathbb{E}\breve{T} - n_i[t]y\right) dy$$

similarly for $e[t]$

$$\mathbb{E}(e[t]) = \left(\sum_{i\in\bar{I}} n_i[t]\right)^2 \int_0^{\mathbb{E}\breve{T}} y^T \mathtt{cov}(\breve{\theta}|\breve{\omega}[t] + \sum_{i=1}^{N} n_i[t]\mathbb{E}\breve{T} - \sum_{i\in\bar{I}} n_i[t]y)dy$$

From Lemma 1, $\texttt{cov}(\breve{\theta}|\breve{\omega}[t]+nx) = \nabla^2\Lambda(\breve{\omega}[t]+nx)$ converges to 0 when $n \to \infty$. It can be further proven from [55] that $\texttt{cov}(\breve{\theta}|\breve{\omega}[t]+nx)$ is convex in n. Hence, $\texttt{cov}(\breve{\theta}|\breve{\omega}[t]+nx)$ monotonically converges to 0. Therefore,

$$\int_0^{\mathbb{E}\breve{T}} y^T \texttt{cov}(\breve{\theta}|\breve{\omega}[t] + \sum_{i=1}^{N} n_i[t]\mathbb{E}\breve{T} - \sum_{i\in\bar{I}} n_i[t]y)dy$$
$$\leq \int_0^{\mathbb{E}\breve{T}} y^T \texttt{cov}\left(\breve{\theta}|\breve{\omega}[t] + n_i[t]\mathbb{E}\breve{T} - n_i[t]y\right) dy$$

Furthermore $(\sum_{i\in\bar{I}} n_i[t])^2 \leq N^2 \max_{i\in\bar{I}}(n_i[t])^2$, we have:

$$\mathbb{E}(e[t]) \leq N^2 \max_{i\in\bar{I}} \mathbb{E}(V_i[t])$$

Because $\mathbb{E}(V_i[t]) < V^*$ for agent $i \in \bar{I}$, we have $\mathbb{E}(e[t]) \leq N^2 V^*$. ■

3.4 Adaptive VoI Based Distributed Sensing (A-VoIDS)

With a static VoI threshold V^*, the communication frequency of VoIDS was shown to decrease over time and the expected error was shown to be bounded by a constant. In particular, at the beginning of the estimation process, agents know little about the parameters of interest, hence new measurements tend to contain more information, so the set of informative agents is larger and there is more communication in the network. In contrast, at later stages of the estimation process, when agents have developed better estimates of the parameters, new measurements are less informative, so agents declare themselves as informative less frequently. While this means that the growth of the communication cost slows down, the error still remains bounded by V^* instead of continuing to decrease. Note that the number of agents declaring themselves as informative depends on V^*. Hence, for a real network that has a fixed communication bandwidth, the VoI threshold needs to be larger in the early stages of estimation to guarantee that the network is not overwhelmed, while in the

later stages of the estimation, V^* must be dynamically reduced in order to guarantee continuous reduction of the estimation error. This implies that there is a dynamic tradeoff between the growth of cost and estimation error, and in a network with fixed communication bandwidth, the tradeoff can be handled by dynamically adjusting the value of V^*.

The Adaptive VoI based Distributed Sensing (A-VoIDS) algorithm discussed in this section provides a way to adaptively adjust the VoI threshold V^* to make most of the available communication bandwidth (defined by preset communication limits in a single time step). Because all agents will get messages from informative agents, all agents know the communication cost in the network at any given time step. Therefore it is possible for agents to update V^* in the same manner without introducing extra communication between them.

Let indicator function $\mathcal{I}_{V_i[t]>V^*[t]}$ denote whether agent i is informative and sends out a message at time t. Let $C[t]$ denote the number of messages sent out at a single time step averaged among a past window of length l:

$$C[t] = \frac{1}{l} \sum_{j=t-l+1}^{t} \sum_{i=1}^{N} \mathcal{I}_{V_i[j]>V^*[j]}. \tag{3.3}$$

Variable $C[t]$ reflects the average step communication cost in a fixed length window. It should be noted that because the VoI of measurements taken by agents is not known a priori, the step cost $C[t]$ is a random variable. If $C[t]$ is too high, the communication cost will grow very rapidly, on the other hand if $C[t]$ is too low, then the error reduces very slowly. Therefore, it is desirable to regulate $C[t]$ around a reference value determined by the available communication bandwidth. A-VoIDS achieves this objective by dynamically adjusting the VoI threshold. In A-VoIDS, each agent compares the incurred $C[t]$ with a desired step-cost c, and adjusts $V^*[t]$ accordingly. If $C[t] < c$, the communication cost is lower than desired, which means that the available communication bandwidth is ill-utilized, hence the algorithm decreases V^* to encourage communication by setting $V^*[t+1] = \gamma_1 V^*[t]$, $0 < \gamma_1 < 1$ (mode 1 of the algorithm). If $C[t] \geq c$, the communication cost is higher than desired, so the algorithm increases

V^* to limit communication by setting $V^*[t+1] = \gamma_2 V^*[t]$, $\gamma_2 > 1$ (mode 2). The above procedure used by the A-VoIDS algorithm is depicted in an algorithmic form in Algorithm 6. In the following theorem it is shown that the A-VoIDS algorithm guarantees that the estimation error asymptotically approaches zero almost surely.

Theorem 3.5 Consider a network of N distributed sensing agents. Assume that the measurements of all agents are i.i.d. drawn from a distribution with static parameters θ_0. Then the estimation error $e[t]$ as defined in (3.2) asymptotically reduces to zero a.s., that is $\lim_{t\to\infty} e[t] \to 0$ a.s. ■

Proof 6 Denote the probability distribution of $V^*[t]$ at time t as $p^t(v)$. At time t, define the probability of being in mode 1 as $\mathbb{P}_1[t|v] = \mathbb{P}(C[t] < c|v)$ and being in mode 2 as $\mathbb{P}_2[t|v] = \mathbb{P}(C[t] \geq c|v)$. From Theorem 3.3, for any given VoI threshold $V^* = v$, the interval between two consecutive updates of any agent i will increase to infinity, hence the probability of sending out a message at a particular time t will approach zero, i.e. $\lim_{t\to\infty} \mathcal{I}_{V_i[t]>v} \to 0$ a.s. Therefore, for a fixed window length l, the average cost $C[t]$ satisfies:

$$\forall v > 0, \quad \lim_{t\to\infty} C[t] = \lim_{t\to\infty} \frac{1}{l} \sum_{j=t-l+1}^{t} \sum_{i=1}^{N} \mathcal{I}_{V_i[j]>v} \to 0 \text{ a.s.}$$

Hence, over time the probability of being in mode 1 will approach 1, while being in mode 2 will approach 0, that is

$$\forall v > 0, \quad \lim_{t\to\infty} \mathbb{P}_1[t|v] = 1, \quad \text{and} \quad \lim_{t\to\infty} \mathbb{P}_2[t|v] = 0$$

For any given $\zeta > 0$, if $V^*[t+1] \geq \zeta$, there are two possibilities, $V^*[t] \geq \frac{\zeta}{\gamma_1}$ and the algorithm falls into mode 1 at t; or $V^*[t] \geq \frac{\zeta}{\gamma_2}$ and the algorithm falls into mode 2 at

Algorithm 6 Adaptive VoI Based Distributed Sensing (A-VoIDS)

initiate hyperparameters $\breve{\omega}[0]$
for t **do**
  for each agent i **do**
    $\breve{\omega}[t] = \breve{\omega}[t-1]$, $C[t] = 0$
    take measurement and update local buffer
    $\breve{S}_i[t] = \breve{S}_i[t-1] + \breve{T}(z_i[t])$
    calculate VoI of current buffer
    $V_i[t] = D_{\text{KL}}\left(p(\theta|\breve{\omega}[t])||p(\theta|\breve{\omega}[t] + \breve{S}_i[t])\right)$
    if $V_i[t] > V^*[t]$ **then**
      broadcasts $\breve{S}_i[t]$
      reset local buffer: $\breve{S}_i[t] = 0$
    end if
    relay received new message $\mathbf{z}_j[t]$ to neighbors
    for each broadcast message $\breve{S}_j[t]$ **do**
      update the global posterior
        $\breve{\omega}[t] = \breve{\omega}[t-1] + \breve{S}_j[t]$
      step communication cost increased by 1
        $C[t] = C[t] + 1$
    end for
    adaptively change $V^*[t]$
    if $C[t] < c$ **then**
      smaller than bound, too little comm
        $V^*[t+1] = \gamma_1 V^*[t]$ $(0 < \gamma_1 < 1)$
    else
      bigger than bound, too much comm
        $V^*[t+1] = \gamma_2 V^*[t]$ $(\gamma_2 > 1)$
    end if
  end for
end for

t, where γ_1, γ_2 are as defined in Algorithm 6. Therefore, we have

$$\begin{aligned}\mathbb{P}(V^*[t+1] \geq \zeta) &= \int_{\zeta}^{\infty} p^{t+1}(v)dv \\ &= \int_{\frac{\zeta}{\gamma_1}}^{\infty} \mathbb{P}_1(t|v)p^t(v)dv + \int_{\frac{\zeta}{\gamma_2}}^{\infty} \mathbb{P}_2(t|v)p^t(v)dv \end{aligned} \tag{3.4}$$

When $t \to \infty$, $\mathbb{P}_1 \to 1$, $\mathbb{P}_2 \to 0$, therefore taking limit w.r.t. time we have

$$\lim_{t\to\infty} \mathbb{P}\left(V^*[t+1] \geq \zeta\right) = \lim_{t\to\infty} \int_{\frac{\zeta}{\gamma_1}}^{\infty} p^t(v)dv = \lim_{t\to\infty} \mathbb{P}\left(V^*[t] \geq \frac{\zeta}{\gamma_1}\right)$$

Move two limits to the same side of the equation, we have:

$$\lim_{t\to\infty} \mathbb{P}\left(V^*[t+1] \geq \zeta\right) - \mathbb{P}\left(V^*[t] \geq \frac{\zeta}{\gamma_1}\right) = 0. \tag{3.5}$$

Noting that $[t+\tau, t] = [t+\tau, t+\tau-1, \cdots, t+1, t]$, (3.5) can be rewritten by adding and subtracting intermediate terms $\mathbb{P}\left(V^*[t+i] \geq \frac{\zeta}{\gamma_1^{\tau-i}}\right)$, $i = 1, \cdots, \tau$.

$$\begin{aligned}&\lim_{t\to\infty} \mathbb{P}\left(V^*[t+\tau] \geq \zeta\right) - \mathbb{P}\left(V^*[t] \geq \frac{\zeta}{\gamma_1^{\tau}}\right) \\ &\qquad = \lim_{t\to\infty} \left\{\mathbb{P}\left(V^*[t+\tau] \geq \zeta\right) - \mathbb{P}\left(V^*[t+\tau-1] \geq \frac{\zeta}{\gamma_1}\right)\right\} \\ &\qquad + \cdots \\ &\qquad + \lim_{t\to\infty} \left\{\mathbb{P}\left(V^*[t+2] \geq \frac{\zeta}{\gamma_1^{\tau-2}}\right) - \mathbb{P}\left(V^*[t+1] \geq \frac{\zeta}{\gamma_1^{\tau-1}}\right)\right\} \\ &\qquad + \lim_{t\to\infty} \left\{\mathbb{P}\left(V^*[t+1] \geq \frac{\zeta}{\gamma_1^{\tau-1}}\right) - \mathbb{P}\left(V^*[t] \geq \frac{\zeta}{\gamma_1^{\tau}}\right)\right\}\end{aligned}$$

apply (3.5) to each of the limits

$$= 0 + 0 + \cdots + 0 = 0$$

Now letting $\tau \to \infty$ we obtain:

$$\lim_{\tau\to\infty} \lim_{t\to\infty} \mathbb{P}\left(V^*[t+\tau] \geq \zeta\right) - \mathbb{P}\left(V^*[t] \geq \frac{\zeta}{\gamma_1^\tau}\right) = \lim_{\tau\to\infty} 0$$

therefore,

$$\lim_{t\to\infty} \mathbb{P}\left(V^*[t] \geq \zeta\right) - \mathbb{P}\left(V^*[t] \geq \infty\right) = 0$$

Because by definition of probability measures $\mathbb{P}(V^*[t] \geq \infty) = 0$, hence we have:

$$\forall \zeta, \quad \lim_{t\to\infty} \mathbb{P}\left(V^*[t] \geq \zeta\right) = 0$$

Therefore, $\lim_{t\to\infty} V^*[t] \to 0$ a.s. From Theorem 3.4, $\lim_{t\to\infty} \mathbb{E}(e[t]) \to 0$ a.s. Since $e[t] \geq 0$, it follows that $\lim_{t\to\infty} e[t] \to 0$ a.s. ■

3.5 Comparison of Performance

Table 3.1 compares the algorithms discussed so far. For each algorithm, it is listed how the communication cost scales with network size and the asymptotic error performance.

Table 3.1: Sensing Algorithm Performance Summary

Algorithm	Cost in a step	Error
Full Relay	N^2	0
HPC	N	converges to 0
Random Broadcast	ϵN^2, ϵ tunable	random; slowlyconverges to 0
VoIDS	converge to 0	bounded
A-VoIDS	cN, c tunable	quickly convergesto 0

It should be noticed the algorithms listed in Table 3.1 are all limited to static parameter scenarios. Next section 3.6 will discuss how the performances will change with dynamic parameters.

3.6 Dynamic VoI Based Distributed Sensing (Dynamic-VoIDS)

In both VoIDS and A-VoIDS, the parameters to be estimated are assumed to be constant. When there are new measurements, agents update local hyperparameters by adding new terms to sufficient statistics, as shown in equation:

$$\breve{S}_i[t] = \breve{S}_i[t-1] + \breve{T}(z_i[t]) \tag{3.6}$$

For all agents, sufficient statistics $\breve{T}()$ of old measurements have the same weight as that of new measurements in hyperparameters $\breve{S}()$. It has been proven that the communication cost (VoIDS) or estimation error (A-VoIDS) asymptotically converge to 0 when the parameters are static.

However, the parameters we are interested in are often dynamic. For instance, in tracking problems, the position and velocity of moving targets always change with time. In these scenarios, VoIDS and A-VoIDS fail to capture the change in parameters as new information is overwhelmed by old information.

To capture dynamics parameters, more focus should be put on new measurements, because new measurement can better reflect the latest value of the parameters we are estimating. Forgetting factor is a widely used technique in estimation/control theories and Marcov decision process to weight information in different stages, [72, 73]. Similarly, a forgetting factor α $(0 < \alpha < 1)$ is also introduced here to do Bayesian update on hyperparameters:

$$\breve{S}_i[t] = \alpha\breve{S}_i[t-1] + \breve{T}(z_i[t]) \tag{3.7}$$

The algorithm is depicted in an algorithmic form in Algorithm 7.

With this discounting factor, information from old measurements will gradually die away and information from new measurements will become dominant. Therefore, the estimation of parameters will gradually convert to new values in a dynamic

Algorithm 7 Dynamic VoI-realized Distributed Sensing (Dynamic-VoIDS)

initiate hyperparameters $\breve{\omega}[0]$
for t **do**
  for each agent i **do**
    $\breve{\omega}[t] = \breve{\omega}[t-1]$
    take measurement and update local buffer
    $\breve{S}_i[t] = \alpha \breve{S}_i[t-1] + \breve{T}(z_i[t])$
    calculate VoI of current buffer
    $V_i[t] = D_{\mathtt{KL}}\left(p(\theta|\breve{\omega}[t])||p(\theta|\breve{\omega}[t] + \breve{S}_i[t])\right)$
    if $V_i[t] > V^*[t]$ **then**
      broadcasts $\breve{S}_i[t]$
      reset local buffer: $\breve{S}_i[t] = 0$
    end if
    relay received new message $\mathbf{z}_j[t]$ to neighbors
    for each broadcast message $\breve{S}_j[t]$ **do**
      update the global posterior
        $\breve{\omega}[t] = \alpha \breve{\omega}[t-1] + \breve{S}_j[t]$
    end for
  end for
end for

scenarios. More specifically, forgetting factor α reflects the speed of forgetting old information from new information. With smaller α, old information is forgotten faster and algorithms converts to new values faster. With biger α, old information is forgotten slower and algorithms converts to new values slower. On the other hand, when α is smaller, algorithm is forgetting information fast so new measurements tend to be more informative and communication cost tends to increase quickly. When α is bigger, algorithm is forgetting information slower so new measurements tend to be less informative and communication cost tends to increase lower. Hence, the choice of forgetting factor α should reflect the trade-off between the parameter tracking speed and communication cost.

3.7 Summary

Chapter 3 discussed how to efficiently use limited communication resources to collaboratively sensing parameters. In particular, VoI based Distributed Sensing (VoIDS) algorithm is developed, in which agents communicate with neighbors only when value of their own measurements exceeds a predefined threshold. However, it was shown there was a trade-off between the communication cost and long term estimation error. Therefore, Adaptive VoIDS is developed which can dynamically change VoI thresholds to achieve a balance between communication cost and estimation error. Asymptotic theoretical bounds of VoIDS and A-VoIDS are given in terms of cost and error. To further deal with dynamically changing parameters, Dynamic-VoIDS is developed.

Next Chapter 4 will discuss how to use the sensing abilities of agents to benefit the overall mission returns of the system.

Chapter 4

Value of Information Based Distributed Planning

This chapter develops the algorithms for distributed planing problems. In particular, robust planning algorithms are first described which use mathematical programming techniques to assign agents to different missions in uncertain environment. When there are dedicate agents that can take measurements and reduce uncertainties, an algorithm that can evaluate the information gathered by the increment of mission returns is developed. Finally, to deal with various attributes of missions and agents in a scalable way, an algorithm based on a Consensus-Based Bundle Algorithm (CBBA) is introduced to assign heterogeneous agents with additional system constraints.

4.1 Robust Planning

In a typical agent-mission assignment problem, the goal is to allocate a team of n agents to N targets to perform tasks with the objective of maximizing the returns. The team accrues reward associated with the targets to which agents are assigned. If no agents are assigned to targets, then the team receives a return of 0. This problem becomes challenging when the returns are uncertain.

Assume for now the time is discrete and indexed by an integer $t \in \mathbb{N}$. At time t, let vector $\mathbf{C}_t = [C_{1,t} \cdots C_{N,t}]^T$ denotes the rewards got by finishing tasks associated

with N individual target. Since the reward is uncertain, $\mathbf{C}_t$ is a vector of random variables instead of deterministic values. Binary variable vector $\mathbf{x}_t = [x_{1,t} \cdots x_{N,t}]^T$ where $x_{i,t} \in \{0, 1\}$, denotes whether there is an agent assigned to each task i at time t. Further let $f(C) \in \mathcal{R}$ denote a functional mapping from the probability distribution of reward C to a score, then the planning problem can be recast as an integer programming problem:

$$\begin{aligned} \max \quad & \sum_{i=1}^{N} f(C_{i,t}) x_{i,t} \\ \text{s.t.} \quad & \sum_{i=1}^{N} x_{i,t} \leq n, \quad \in \{0, 1\} \end{aligned} \tag{4.1}$$

Score function $f(C)$ should capture two basic properties of reward C, the potential return and the uncertainty in it. Higher return and lower uncertainty should lead to larger scores $f(C)$. Different $f(C)$ leads to different perspectives on trade-off between potential return and uncertainty.

For example, the basic stochastic programming formulation uses expected reward $f(C) = \mathbb{E}(C)$ [37]:

$$\begin{aligned} \max \quad & \sum_{i=1}^{N} \mathbb{E}(C_{i,t}) x_{i,t} \\ \text{s.t.} \quad & \sum_{i=1}^{N} x_{i,t} \leq n, \quad x_{i,t} \in \{0, 1\} \end{aligned} \tag{4.2}$$

This model only incorporates first moment information, which can potentially assign agents to targets with large uncertainty.

When C is bounded, $f(C)$ can account for uncertainty by taking the worst case value $f(C) = \min C$: [74]

$$\begin{aligned} \max_{x} \quad & \sum_{i=1}^{N} \left(\min_{C_{i,t}} C_{i,t} \right) x_{i,t} \\ \text{s.t.} \quad & \sum_{i=1}^{N} x_{i,t} \leq n, \quad x_{i,t} \in \{0, 1\} \end{aligned} \tag{4.3}$$

This model can lead to very conservative plans because it is based on the worst case scenario.

Bertuccelli [35] proposed a planning formulation that can make a trade-off between expected reward and uncertainty. He assumes the distribution of reward is Gaussian: $C \sim \mathcal{N}(\bar{c}, \sigma^2)$, and score function is $f(C) = \bar{c} - \mu\sigma$ in his framework. Compared to the expected value formulation (4.2) and the worst case formulation (4.3), this formulation strikes a balance between the expected return $\bar{c}$ and the uncertainty σ by picking different μ.

$$\begin{aligned} \max_{x} \quad & (\bar{c}_{i,t} - \mu\sigma_{i,t})x_{i,t} \\ \text{s.t.} \quad & \sum_{i=1}^{N} x_{i,t} \leq n, \quad x_{i,t} \in \{0,1\} \end{aligned} \tag{4.4}$$

However, the assumption of Gaussian could be limiting in many cases. For example, when the uncertainty of the reward comes from the classification of the target as shown in Figure 1-2, a Gaussian assumption fails to capture the uncertainty features.

The chance-constrained approach is another way to formulate robust planning problems [75, 76, 77, 78]. This approach guarantees the worst case return but allows a certain risk. As shown below, for each target i, the planner maximizes the worst case return $\hat{c}_{i,t}$, but allows a risk ϵ that the return is below $\hat{c}_{i,t}$.

$$\begin{aligned} \max_{x} \quad & \sum_{i=1}^{N} \hat{c}_{i,t} x_{i,t} \\ \text{s.t.} \quad & \mathbf{P}(C_{i,t} < \hat{c}_{i,t}) \leq \epsilon \\ & \sum_{i=1}^{N} x_{i,t} \leq n, \quad x_{i,t} \in \{0,1\} \end{aligned} \tag{4.5}$$

By picking different risk threshold ϵ, this algorithm can also strike a balance between the potential return and uncertainty in the target, and it does not constrain the form of uncertainties.

4.2 Robust Planning of Heterogeneous Agents

The formulations mentioned in Section 4.1 does not account for the case where agents are heterogeneous. When dedicated exploration agents exist, the team performance can be improved by assigning them to reduce the uncertainty in the rewards. This section discusses how the presence of exploration agents change the optimal planning problem.

4.2.1 Exploration-Tasking Decoupled Planning

Traditionally exploration activities and task assignments are taken as two separately planning problems. Exploration agents and tasking agents are allocated in two planners with different score functions, then team assignments are got by simply combining the two allocation results.

Assume there are n tasking agents and m exploration agents. Binary variable vectors $\mathbf{x}_t = [x_{1,t} \ \cdots \ x_{N,t}]^T$, $\mathbf{y}_t = [y_{1,t} \ \cdots \ y_{N,t}]^T$ denote whether a tasking agent or a exploration agent is assigned to each of N targets at time t. Denote the possible measurements of each target at t as $\mathbf{z_t} = [z_{1,t} \ \cdots \ z_{N,t}]^T$.

The assignments of tasking agents can be formed as the following robust planning problem (4.6):

$$\begin{aligned} \max \quad & \sum_{i=1}^{N} f(C_{i,t}) x_{i,t} \\ \text{s.t.} \quad & \sum_{i=1}^{N} x_{i,t} \leq n, \quad \in \{0,1\} \end{aligned} \tag{4.6}$$

The exploration agents are allocated by solving another planning problem, with a different score function $g(C_{i,t})$ (4.7). Function $g(C_{i,t})$ should capture the uncertainty

of reward by exploring the target.

$$\begin{aligned} \max \quad & \sum_{i=1}^{N} g(C_{i,t}) y_{i,t} \\ \text{s.t.} \quad & \sum_{i=1}^{N} y_{i,t} \leq m \quad y_{i,t} \in \{0,1\} \end{aligned} \tag{4.7}$$

Equations (4.6) and (4.7) together give a complete plan of assignments of tasking agents and exploration agents.

4.2.2 Exploration-Tasking Coupled Planning

In the decoupled algorithm, tasking agents and exploration agents are allocated separately with their corresponding score functions. In particular, tasking agents are assigned to targets with higher scores $f(C_{i,t})$ while exploration agents are assigned to targets with higher uncertainty $g(C_{i,t})$. It is often the case that these two targets sets do not have much overlap. If this is the case, tasking agents do not benefit much from the exploration as exploration agents are assigned to a different set of targets. Bertuccelli proposed a method to couple the assignments of tasking agents and exploration agents in a universal planner [35]. However, his approach is only limited to Gaussian uncertainties. A similar but more general approach is proposed here, which couple the value of doing exploration with the increase in mission returns.

In section 2.3.2, it is discussed how to evaluate the exploration activities by increment in mission returns. Let $f(C_{i,t})$ denote score of doing tasks at target i at time t. Then after doing exploration, the expected posterior score will become $\mathbb{E}f(C_{i,t+1})$, defined in equation (2.11). The system accrue reward only when a tasking agent is assigned to perform mission at a target. The score is $\mathbb{E}f(C_{i,t+1})$ if an exploration agent is assigned to take measurements of the target, and is $f(C_{i,t})$ if no exploration agents are assigned. This idea gives the following exploration-tasking coupled planning algorithms:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{N} \mathbb{E}f(C_{i,t+1})x_{i,t}y_{i,t} + f(C_{i,t})x_{i,t}(1-y_{i,t}) \\
\text{s.t.} \quad & \sum_{i=1}^{N} y_{i,t} \leq m, \quad y_{i,t} \in \{0,1\} \\
& \sum_{i=1}^{N} x_{i,t} \leq n, \quad x_{i,t} \in \{0,1\}
\end{aligned} \tag{4.8}
$$

Binary variables $x_{i,t}$ denote whether a tasking agent is assigned to target i at time t and $y_{i,t}$ denote whether an exploration agent is assigned to target i. The reward at target i is non-zero only when $x_{i,t} = 1$. The score function is $f(C_{i,t})$ if $y_{i,t} = 0$, and is $\mathbb{E}f(C_{i,t+1})$ if $y_{i,t} = 1$.

In this framework, exploration agents do not have separate goals, their value is coupled into task returns. Exploration agents can add value only when they are assigned to targets that tasking agents are performing missions on. Therefore, this framework lead to cooperation between heterogeneous agents where agents are assigned to doing tasks as well as reduce uncertainties to increase returns of tasks at the same time.

4.3 Distributed Planning with System Dynamics

The frameworks discussed in Section 4.2 are centralized and assume that the agents can get to the target within the time step once being assigned to it, and exploration agents can always get measurements done before tasking agent perform tasks. However, the system is often dynamic and there are physical constraints such as positions and task durations of targets, positions/velocities of agents. These dynamics and constraints can affect the assignments results. For example, in order for the tasking agent to benefit from exploration at a target, the exploration agent should finish taking measurements before the corresponding tasking agent starts doing tasks at that target. Mathematical programming based approaches cannot deal with these

constraints easily, because the increase in number of agents and tasks in dynamics systems leads to combinational increase in candidate solutions and becomes intractable quickly. It is also beneficial for the framework to be decentralized to get benefits as robustness and scalability. This section discusses how a polynomial decentralized algorithm, Consensus-Based Bundle Algorithm (CBBA) can be utilized to solve the planning problems discussed in section 4.2.

4.3.1 Consensus-Based Bundle Algorithm (CBBA)

Consensus-Based Bundle Algorithm (CBBA) is a distributed agreement protocol where agents bid on tasks to decide the assignments. It can well handle system dynamics, and provides provably good approximate solutions for multi-agent multi-task allocation problems over networks of agents [79, 80].

CBBA consists of iterations between two phases: a bundle building phase where each vehicle greedily generates an ordered bundle of tasks, and a consensus phase where conflicting assignments are identified and resolved through local communication between neighboring agents. There are several core features of CBBA that can be exploited to develop an efficient planning mechanism for heterogeneous teams. First, CBBA is a decentralized decision architecture, which is a necessity for planning over large teams due to the increasing communication and computation overhead required for centralized planning. Second, the complexity of CBBA is polynomial in the number of tasks and vehicles. Third, various design objectives, agent models, and constraints can be incorporated by defining appropriate scoring functions.

Table 4.1 shows the features of CBBA that will be used in this work. The input of CBBA are agent and task attributes. All agents and tasks are identified by unique IDs. An agent is defined by its location and speed. A task is defined by the location, value, start and duration of this task. The output of CBBA is a list of assignments. Each assignment gives a pair of agent ID and task ID and the expected time the agent starts doing this task.

Table 4.1: Consensus-Based Bundle Algorithm (CBBA)

Input

Agent	agent ID agent location speed
Task	task ID task location start time duration max score

Output

Assignment	agent ID task ID stat time

Algorithm 8 Decoupled Distributed Planning

1: compute task score functions $f(C_i)$, $i = 1, \cdots, N$
2: call CBBA to assign tasking agents
3: agents: tasking agents
4: tasks: value $f(C_i)$, $i = 1, \cdots, N$
5: compute task uncertainty functions $g(C_i)$, $i = 1, \cdots, N$
6: call CBBA to assign exploration agents
7: agents: exploration agents
8: tasks: value $g(C_i)$, $i = 1, \cdots, N$

4.3.2 Value of Information Based Planning based on CBBA

In decoupled approach, exploration and tasking agents are allocated separately. Given agents, tasks and distribution of task rewards, decoupled approach computes the score functions $f()$ and $g()$ separately, then call CBBA twice to get the assignments of exploration agents and tasking agents. The algorithm is shown in Algorithm 8.

Coupled approach iterates between assignments of tasking agents and exploration agents to search for the best assignment. First define a candidate plan $\mathcal{P}$ as an evaluation of tasks, whose score is either $f()$ or $\mathbb{E}[f()]$.

$$\mathcal{P} = \{c_1, c_2, \cdots, c_N\} \quad c_i = f(C_i) \text{ or } \mathbb{E}[f(C_i)] \tag{4.9}$$

Algorithm 9 VoI Based Coupled Distributed Planning

compute task score functions $f(C_i)$, $i = 1, \cdots, N$
initialize candidate solution queue Q
  $Q = \{\mathcal{P}_0\}$, $\mathcal{P}_0 = \{\max(\ f(C_i),\ \mathbb{E}[f(C_i)]\ ), i = 1, \cdots, N\}$
while tasking agent requirement not satisfied **do**
  $\mathcal{P}$ = nextBestCandidate(Q)
  call CBBA to assign tasking agents
    $\mathcal{A}_T$ = CBBA($\mathcal{P}, agents, tasks$)
  call CBBA to assign exploration agents
    agents: exploration agents
    $\mathcal{A}_E$ = CBBA($\mathcal{A}_T, agents, tasks$)
  check whether tasking agent requirement satisfied
    checkMatch($\mathcal{A}_T, \mathcal{A}_E$)
end while
return $\mathcal{A}_T$, $\mathcal{A}_E$

Given the best candidate plan $\mathcal{P}$ so far, CBBA is called to compute the assignment of tasking agents $\mathcal{A}_T$. The tasking agent assignment $\mathcal{A}_T$ is then passed to exploration agents. Exploration agents get positive scores at a target only when tasking agents is doing tasks there (c_i is in the list of $\mathcal{A}_T$) and require exploration to be done first ($c_i = \mathbb{E}[f(C_i)]$). CBBA is then called to compute the assignment exploration agents $\mathcal{A}_E$. If exploration agents can finish all the exploration requirements that tasking agents have proposed, then the algorithm return the solution $\mathcal{A}_T, \mathcal{A}_E$. Otherwise, tasking agents will work on the next candidate assignment with highest score. All candidate plans are stored in a priority queue Q. The algorithms is outlined in Algorithm 9.

4.4 Summary

This chapter discussed distributed planning algorithms of multi-agent system with heterogeneous abilities. In particular, the VoI based Coupled Planning algorithms directly use the increase in mission return as metric of information gathering activities. The performance of algorithms discussed in this chapter as well as Chapter 3 will be studied numerically by both simulated and real-world data in next chapter.

Chapter 5

Numerical Evaluation

This chapter discusses the performance of algorithms developed in earlier chapters by numerical studies. In the first section, the performance of distributed sensing algorithms is compared based on the required communication cost incurred and error to the centralized Bayesian estimate (which is assumed to be the truth). Both simulated data and a real dataset are used to compare the performance of VoI based algorithms with existing distributed sensing algorithms. The second sections first compare the performance of decoupled and VoI aware coupled distributed planning algorithms through simulation, then states a physical testbed to compare the algorithms in real world scenarios.

5.1 Evaluation of Distributed Sensing Algorithms

In this section, simulated data and a real dataset are used to compare the performance of VoI based algorithms with existing distributed sensing algorithms (Full Relay, Random Broadcast, and HPC, see Section 2 for details) in terms of the communication cost incurred and the error to the centralized Bayesian estimate (which is assumed to be the truth).

5.1.1 Evaluation Using Simulated Dataset

The presented simulation considers a group of collaborative agents estimating the Poisson arrival rate λ of an entity. The prior distribution of λ is chosen to be a Gamma distribution $\Gamma(\alpha, \beta)$, which is conjugate to Poisson. The Poisson and Gamma distributions are given in (5.1) and (5.2) respectively:

$$p(z|\lambda) = \frac{(\lambda)^z e^{-\lambda}}{z!} \tag{5.1}$$

$$p(\lambda|\alpha, \beta) = \frac{\beta^\alpha \lambda^{\alpha-1} e^{-\beta\lambda}}{\Gamma(\alpha)} \tag{5.2}$$

This conjugacy results a closed-form update of hyperparameters when a measurement z is taken:

$$\begin{aligned} \alpha &= \alpha + z \\ \beta &= \beta + 1 \end{aligned} \tag{5.3}$$

The total number of agents in the network is set to be 100. At each time step t, every agent i takes one measurement $z_i[t] \sim Poi(\lambda_i)$. The local arrival rate parameters λ_i are biased from the true global value $\lambda = 5$ with uniform noise: $\lambda_i \sim U(4, 6)$. For VoIDS, the VoI threshold are chosen to be $V^* = 0.02, 0.1, 0.5$ respectively. For A-VoIDS, the parameters are set to $\gamma_1 = 0.97$, $\gamma_2 = 1.01$ $l = 30$, $V^*[0] = 0.5$, and two communication bandwidths are tested $c = 0.10, 0.05$.

Figures 5-1 and 5-2 show the cumulative cost (the sum of all step costs up to current time) and the error to centralized Bayesian estimate of Full Relay, HPC, Random Broadcast, VoIDS and A-VoIDS algorithms (error shown in Figure 5-2 is smoothed over a window with $l = 30$).

Since the step communication costs of consensus-based approach (HPC), Full Relay, and Random Broadcast are constant (see Section 2), the cumulative costs of these algorithms increase linearly. The cost of HPC is significantly less than Full Relay, the cost of Random Broadcast is less than HPC for the chosen probability of communicating/censoring (see Section 3.1.3). Full Relay converges to the centralized

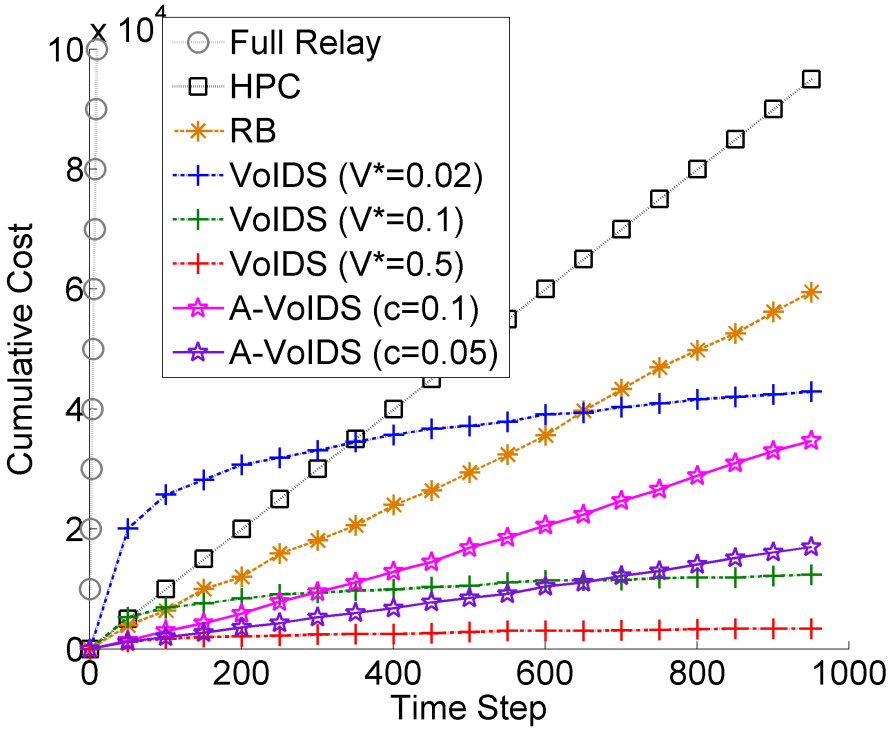


Figure 5-1: Comparison of cumulative cost. The cost of HPC and Random Broadcast is less than Full Relay. The cost of VoIDS can be more than HPC at the beginning, but levels off quickly. Cost of A-VoIDS grows continuously, but its rate of growth is controlled.

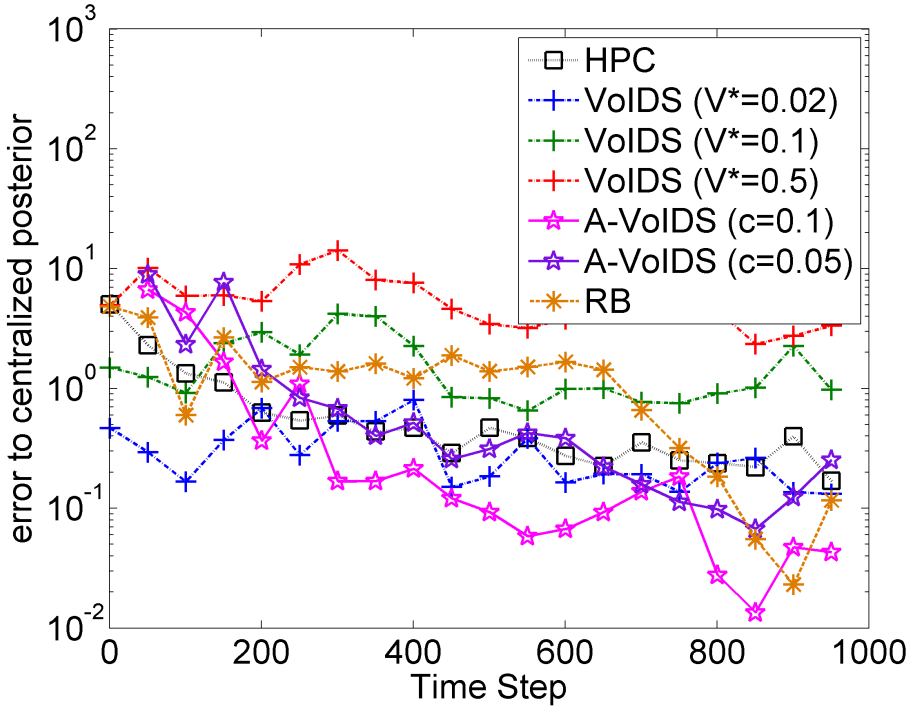


Figure 5-2: Comparison of error to centralized posterior. Full Relay converge to the correct answer immediately and has zero error (not shown). VoIDS is not able to continuously reduce the error, while HPC, Random Broadcast and A-VoIDS can.

Bayesian estimation immediately and has zero error, however, its communication cost is the highest of all the algorithms. Both HPC and Random Broadcast continuously reduce their estimation error, implying asymptotic convergence.

As proven by Theorem 3.3, the cumulative cost of VoIDS tends to grow quickly at the beginning (cost of VoIDS with lower broadcast threshold V^* can be higher than the cost of HPC and Random Broadcast at the beginning), however it levels off gradually as the step cost (3.3) approaches zero. On the other hand, VoIDS has a steady state error because the fixed V^* threshold prevents communication after some time into the simulation, thus cannot further reduce the error. In particular, with a high V^* threshold, agents have less communication cost but higher estimation error, and with a lower V^* threshold, agents have lower estimation error but higher communication cost. This indicates a dynamic tradeoff between communication cost and estimation error.

A-VoIDS (Algorithm 6) strikes a better balance between communication cost and estimation error. The cumulative cost of A-VoIDS also increases linearly, but the rate of growth can be tuned via c to reflect the available communication bandwidth. The cost of A-VoIDS is observed to be less than HPC for the chosen parameters. The evolution of V^* for A-VoIDS ($c = 0.10$) is shown in Figure 5-2, and it can be seen that V^* drops to zero as shown by Theorem 3.5. This indicates that unlike VoIDS, A-VoIDS tends to asymptotically reduce the error, since the estimation error is bounded above by V^*.

The performance of the algorithms discussed is compared in Figure 5-4 in cost-error coordinates. The horizontal axis represents the final cost at the end of the simulation and the vertical axis represents the average error to the centralized estimate of the hyperparameters (the centralized estimate converges to the correct hyperparameters) in the last 300 time steps. An ideal algorithm would be situated in the bottom-left corner of that graph, with low error and low communication cost. HPC is situated in the bottom-right corner, with low error but high communication cost. VoIDS with bigger V^* thresholds (e.g., $V^* = 0.5$) is in the top-left corner, with low communication cost (because the agents do not declare themselves as informative

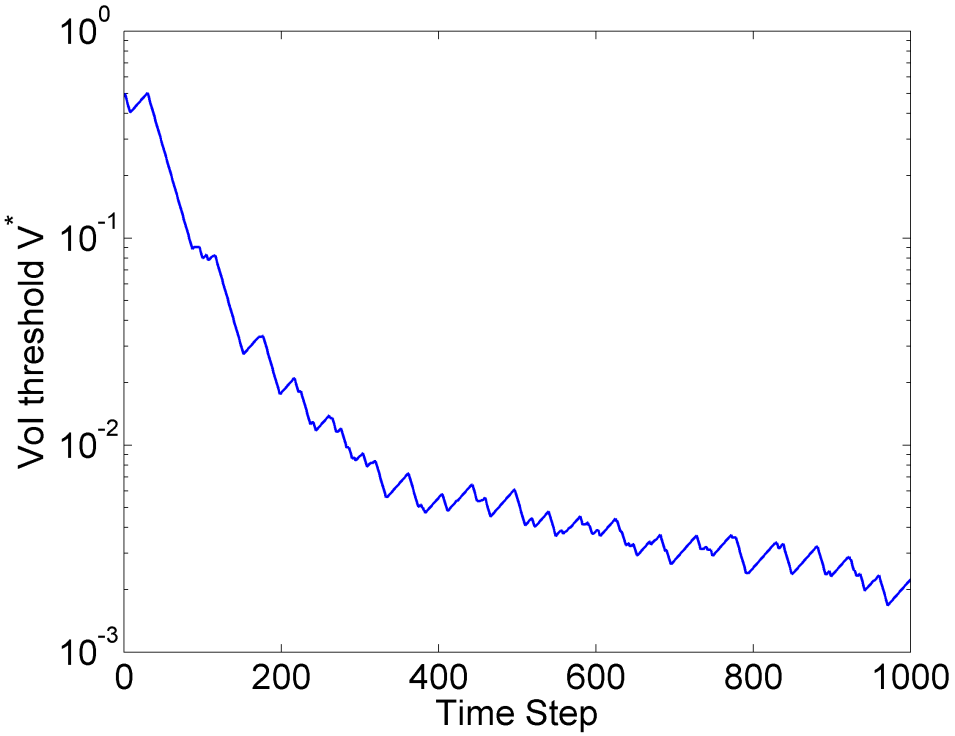


Figure 5-3: VoI threshold V^* in A-VoIDS ($c = 0.10$) asymptotically drops to 0, as postulated by Theorem 3.5. Since the error is bounded by V^* (Theorem 3.4), the estimation error also approaches 0.

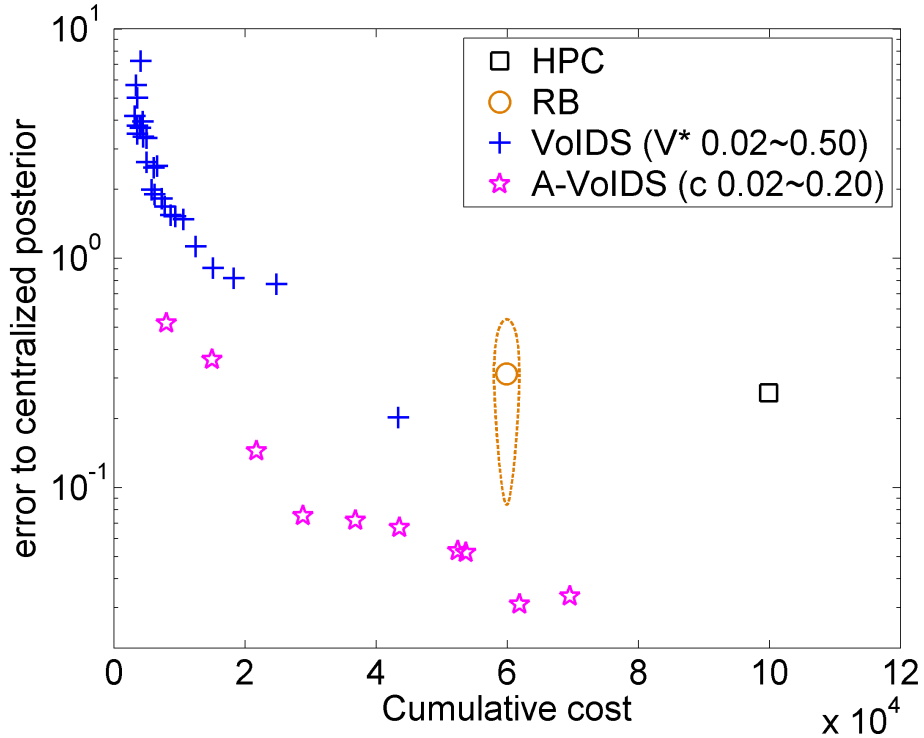


Figure 5-4: Error to Centralized Posterior vs Cumulative Cost by simulated data. A-VoIDS is closest to the left bottom corner, indicating it has less communication cost and less error.

easily) but high error. When V^* is set to lower values, the algorithm results in lower error but higher communication cost. The Random Broadcast algorithm does a trade off between cost and error, however the performance is not always consistent due to the randomness in which agents broadcast the measurement. The bronze circle represents the average performance of 100 runs, and the dashed circle around it indicates one standard deviation of the cost and error. Several instances of A-VoIDS are plotted for different values of c, and increasing c will result in increased communication cost but lower estimation error. The general trend observed is that A-VoIDS dominates the bottom left half of the figure when compared to other algorithms. This indicates that A-VoIDS is capable of achieving excellent estimates without incurring high communication cost.

5.1.2 Evaluation Using Real Dataset

The VoIDS and A-VoIDS algorithms are further evaluated using a real dataset that has been used to analyze distributed sensing algorithms [81, 30]. In this dataset, 54 sensors distributed in the Intel Berkeley Research lab collect timestamped information such as humidity, temperature, light, and voltage values every half minute [81, 30]. The sensor layout is shown in Figure 5-5. This dataset reflects real effects such as sensor noise, sensor bias, and time varying albeit slowly drifting parameters. The temperature data was selected for evaluating the algorithm. Over shorter intervals (e.g. an hour), the drift in temperature in a climate-controlled room is typically small (within 0.2°C) and can be assumed to be approximately constant. Therefore a record of about an hour's temperature measurements is selected to evaluate the algorithms. The positive results reported in this section indicate that the algorithms tend to work well even when the parameters to be estimated are slowly varying instead of being constant as is assumed in the theoretical development.

The goal is to collaboratively estimate the the average room temperature denoted by θ. We model the noisy sensor measurements by each sensor using a Gaussian noise model with constant variance, $z \sim \mathcal{N}(\theta, 1)$. Since the conjugate prior of a Gaussian distribution is also Gaussian, the Gaussian sensor noise model allows for a

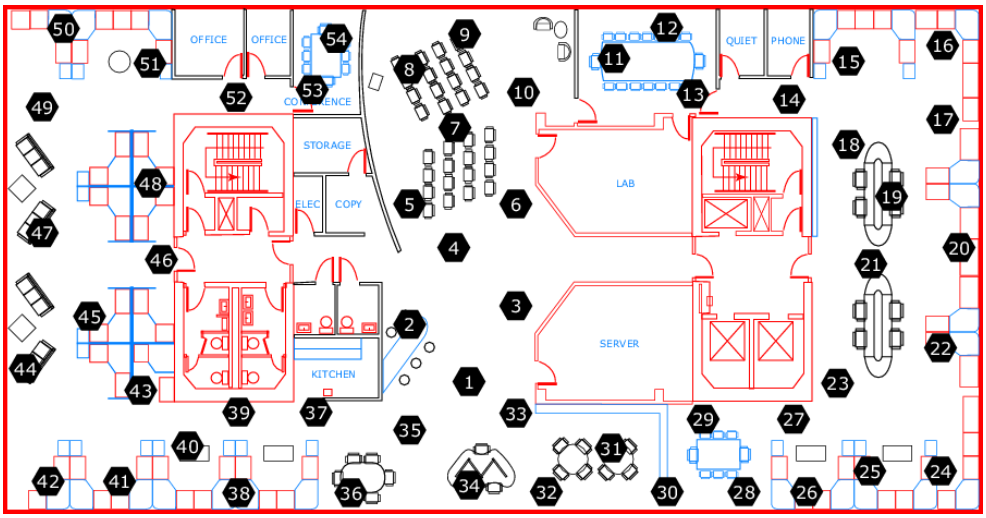


Figure 5-5: Sensor Layout in the Intel Berkeley lab, 54 distributed sensors collect time stamped information such as temperature, humidity, light etc.

closed form update of the hyperparameters. Hence, the average room temperature can be modeled by $\theta \sim \mathcal{N}(\frac{\omega}{\nu}, \frac{1}{\nu})$, where ω, ν are the hyperparameters of the Gaussian conjugate prior. When a new measurement z is taken by a sensor, it updates its hyperparameters as in [67]:

$$\begin{aligned} \omega &= \omega + z, \\ \nu &= \nu + 1. \end{aligned} \tag{5.4}$$

The rate at which the sensors check for or relay messages is a tunable parameter in this scenario, and it can affect the performance of HPC. In particular, increasing this rate tends to increase the speed of HPC error reduction but also increases HPC communication cost by increasing the number of messages sent out. In the presented results, we compare HPC's performance over a range of message communication rates: 1Hz, $\frac{1}{2}$Hz, $\frac{1}{3}$Hz, and $\frac{1}{4}$Hz. Note that the communication rate does not affect the number of messages sent out when sensors are running VoIDS or A-VOIDS.

The performance of different algorithms is compared in Figure 5-6 for different values of communication rate for HPC, V^* for VoIDS, and c for A-VoIDS. At the end of the evaluation run, HPC (1 Hz) results in an estimate with the least error but highest communication cost of all HPC runs. Decreasing the communication

rate decreases HPC communication cost, but the error increases. VoIDS significantly reduces the communication cost compared to HPC, but also has a larger error. The Random Broadcast reduces cost by randomly censoring agents, but its performance has high variance and is no better than VoIDS on average. As before, A-VoIDS with higher values of c are situated closest to the bottom-left corner of the error-cost figure, indicating that A-VoIDS can give an estimate with significantly less communication cost than other algorithms that have similar error.

Figure 5-7 shows the estimated room temperature over time. The horizontal axis represents the time in seconds, and the vertical axis represents the temperature in Celsius. The blue solid line shows the centralized Bayesian estimate. The performance of the Random Broadcast algorithm has a lot of variance. The error between HPC (1Hz) and the centralized estimate drops within 0.1°C within first 500 seconds and keeps decreasing. After 1500 sec, the error is within 0.05°C. VoIDS (with $V^* = 0.1$) estimation error also drops within 0.1°C in 500 sec, but does not further decrease, even after 2000 sec, the error is still as much as 0.1°C. On the other hand, A-VoIDS ($c = 0.1$) starts with larger error than VoIDS ($V^* = 0.1$), but the error quickly decreases, and over time the A-VoIDS error is less than that of HPC.

5.1.3 Evaluation of Dynamic-VoIDS

When parameters are changing with time, the assumption of static parameter becomes invalid. The centralized Bayesian posterior also fails to reflect the true parameter values. In this section, evaluation is still done over the Intel Lab temperature dataset, but a different interval in early morning when the temperature increased dramatically is picked. The mean temperature calculated from all sensor data is assumed to be the truth instead.

The goal of these sensors is still collaboratively estimate the average room temperature θ. The model used is same as that in Section 5.1.2. Both VoIDS and A-VoIDS still assumes the parameter θ is constant and doing estimation accordingly. The dynamic-VoIDS uses a discounting factor α as defined in Equation (3.7) to ac-

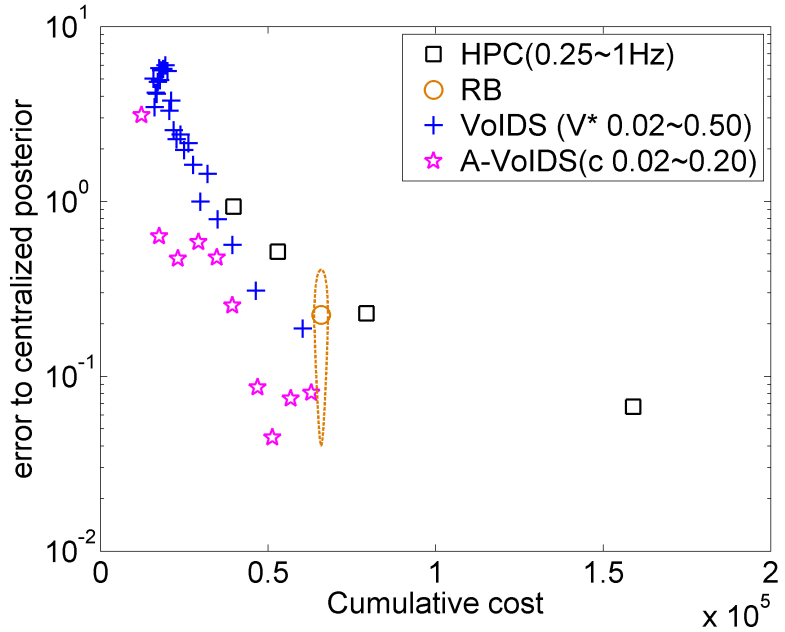


Figure 5-6: Error to centralized posterior vs cumulative communication cost on the Intel dataset [81, 30]. A-VoIDS tends to cluster near bottom left half of the figure, indicating that it can strike an excellent balance between accuracy and communication cost. It can be seen that A-VoIDS outperforms HPC in cost-error coordinates for various values of communication rates.

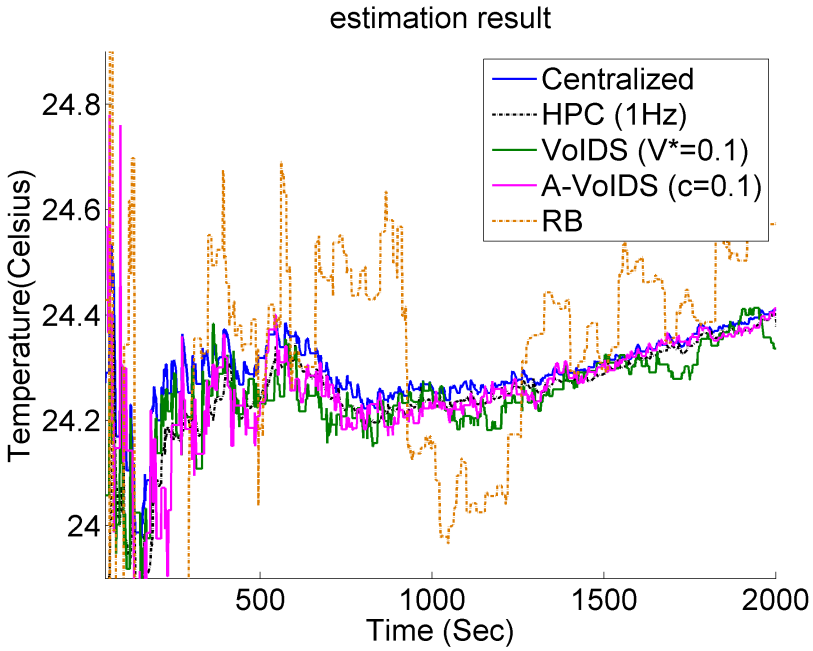


Figure 5-7: Estimated room temperature by different algorithms when new measurements are taken continuously. Centralized estimate is believed to be the truth. RB algorithm has a lot of variance. HPC (1Hz) error drops within 0.1°C within first 500 sec and keeps decreasing to less than 0.05°C. VoIDS ($V^* = 0.1$) estimate also drops within 0.1°C in the first 500 sec, but is unable to decrease further. A-VoIDS ($c = 0.1$) starts with a larger error than VoIDS ($V^* = 0.1$), but drops quickly to within 0.05°C, which is better than that of both HPC and VoIDS.

count for possible dynamics in parameters. Here α is set to be 0.9 to show how the Dynamic-VoIDS works.

Figure 5-8 shows the behavior of VoIDS, A-VoIDS and Dyanmic-VoIDS when the parameters are dynamic. During this interval, the room temperature increases from 16.5°C to 21°C within a period of 20min. Both VoIDS and A-VoIDS can only slowly change towards the new temperature, and are still far away from the correct value even after 50min. On the other hand, Dynamic-VoIDS can respond to the change in temperature quickly and capture the new value in a couple of minutes.

Figure 5-9 shows the cumulative communication cost of VoIDS, A-VoIDS and Dynamic-VoIDS. When the temperature changes, new measurements tend to be more informative, so the cost of VoIDS grows faster than that before the change. A-VoIDS is dynamically changing the VoI threshold therefore is able to constrain the communication cost from growing quickly. The cost of Dynamic-VoIDS is between VoIDS and A-VoIDS after the temperature change. It can correctly capture the dynamics in parameters and effectively use communication resources.

5.2 Evaluation of Distributed Planning Algorithms

This section shows the performance by implementing the framework in Section 4.3 to an assignment problem with categorical reward uncertainties, and presents a hardware platform to further test the algorithms with real world data.

5.2.1 Simulation Setup

The reward uncertainty of a target is modeled by a categorical distribution: reward C of a target takes values in a finite set associated with K categories of target. i.e., $[c^{(1)}, c^{(2)}, \cdots, c^{(K)}]$. For example, when the uncertainty comes from classification of a target, the reward can be modeled by a categorical distribution.

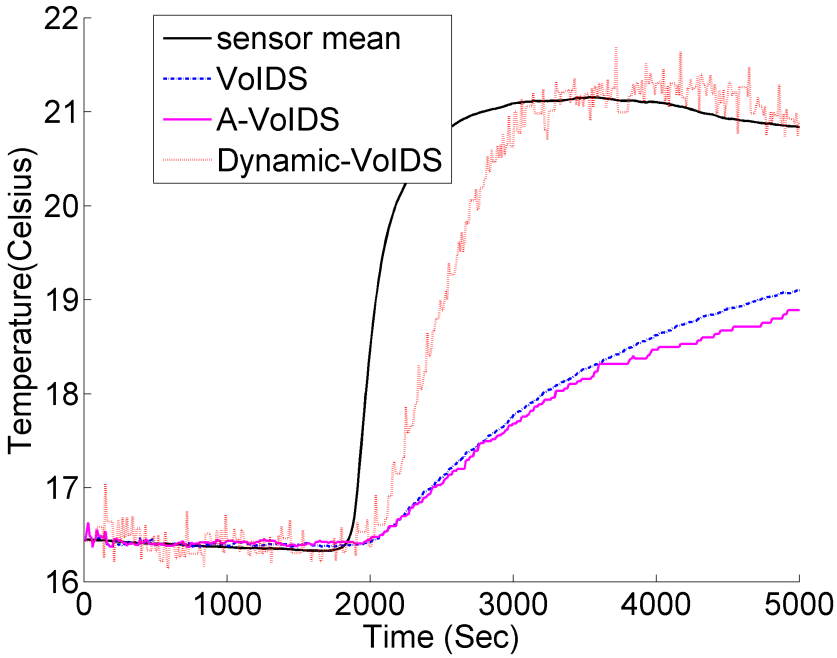


Figure 5-8: Estimated room temperature by different algorithms. The mean of all sensor data is believed to be the truth. When there is a significant change in parameters (around 2000 second), both VoIDS and A-VoIDS estimate make changes slowly and fails to capture the correct estimate. But Dynamic-VoIDS is able to convert to the new parameters quickly and therefore catch the dynamics in parameter change.

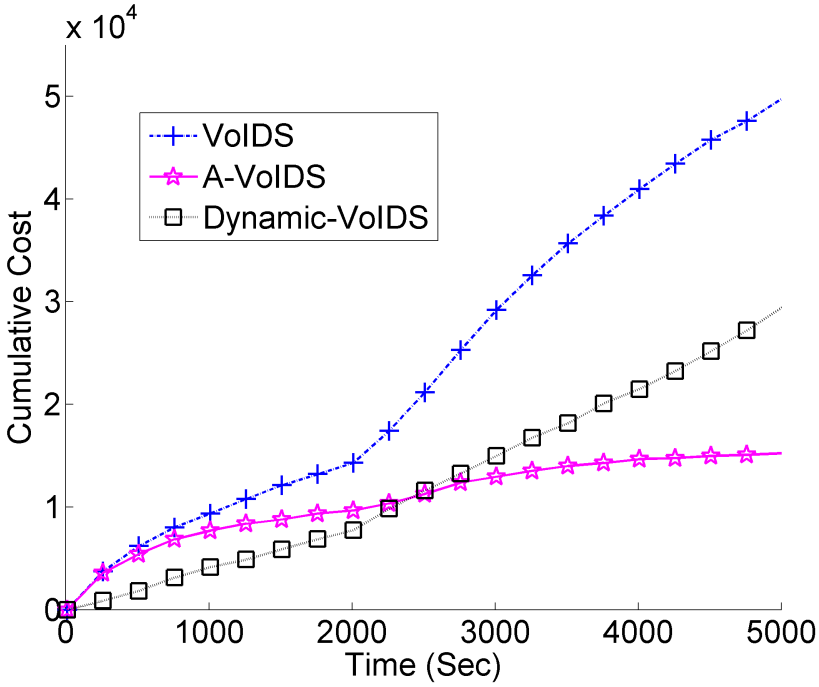


Figure 5-9: Cumulative comm. cost by different distribute sensing algorithms with dynamic parameters.Cost of VoIDS grows faster than A-VoIDS, and cost of Dynamic-VoIDS is between that of VoIDS and A-VoIDS after temperature change.

At time t, the score function $f(C_{i,t})$ is designed in such a way that it guarantees the worst case score $\hat{c}_{i,t}$ but allows a failure chance ϵ [78]:

$$f(C_{i,t}) = \hat{c}_{i,t} \quad \text{s.t. } p(C_{i,t} \geq \hat{c}_{i,t}) \geq 1 - \epsilon \tag{5.5}$$

The measurement model is that exploration agents have a detection rate of γ – if the true score of the target is c_i^k, an exploration agent can get the right classification and the correct value $\mathbf{z}_{i,t} = c_i^k$ with probability γ; and it gets the incorrect values $\mathbf{z}_{i,t} \neq c_i^k$, with equal probability $\frac{1-\gamma}{K-1}$.

$$p(z_{i,t}|c_i^k) = \begin{cases} \gamma & \text{if } \mathbf{z}_{i,t} = c_i^k \\ \dfrac{1-\gamma}{K-1} & \text{otherwise} \end{cases} \tag{5.6}$$

Therefore the expected posterior score function is

$$\mathbb{E}f(C_{i,t+1}) = \mathbb{E}_{\mathbf{z}_{i,t}}(\hat{c}_{i,t+1}) = \sum_{\mathbf{z}_{i,t}} p(\mathbf{z}_{i,t})\hat{c}_{i,t+1}$$

$$p(\mathbf{z}_{i,t}) = \sum_{j=1}^{K} p(\mathbf{z}_{i,t}|c_i^{(j)})p(c_i^{(j)}) \tag{5.7}$$

The uncertainty in classification is measured by entropy:

$$g(C_{i,t}) = \sum_k p(c_i^k)\log(p(c_i^k)) \tag{5.8}$$

During execution stage, when an exploration agent gets a measurement, the probability distribution of the target classification changes, so as the score of tasks. Therefore the system run the algorithm based on the changed distributions and re-plan the assignments.

5.2.2 Simulated Planning Results

Figures 5-10 and 5-11 show an example of the decoupled and coupled planning results. There are 10 targets in this map, each can be one of 3 classifications with score 10,

30 or 100. The histogram beside each target shows the prior probability of the classification for each target. Green triangles represent tasking agents, blue triangles represent exploration gents, and the speed of exploration agents is twice that of tasking agents. Green and blue arrows represent their trajectories respectively. The plots show that in the decoupled approach, tasking agents are assigned to targets based on higher potential reward and exploration agents are assigned to targets that have more uncertainty. It is possible that two groups of agents will end up heading to different sets of targets; if this happens, then the exploration does not help reduce the uncertainties in targets assigned to tasking agents. On the other hand, in the VoI based coupled case, all the exploration agents are paired up with the tasking agents, so exploration agents can always help to reduce the uncertainty of tasks assigned to tasking agents.

During the execution, when an exploration agent takes a measurement of a target, the probability of detecting the right classification is set to be 0.9, and the probability of getting the two other wrong classifications is set to be 0.05. Scenarios with different number of targets are considered. For each scenario, 100 Monte Carlo simulations are run. Three approaches are tested (See Section 4.2 for more detail): in the first approach, the system does not use the exploration capabilities and only tasking agents will be allocated based on prior information about targets; in the second approach, exploration and tasking agents are assigned in a decoupled way; and in the third approach, exploration and tasking agents are assigned in the VoI aware coupled planning way. The performance of these three approaches are compared in Figure 5-12.

The squares show the average score obtained by three different approaches with different number of targets. The vertical lines associated with each square represent the standard deviation of scores associated with each target number. In the tasking agent only case, the system does not utilize its exploration capabilities to reduce the uncertainty about the targets, thus it always gets the lowest score compared with the others. This approach can act as a baseline to show how the exploration agents can improve the performance. The coupled approach gets the highest score

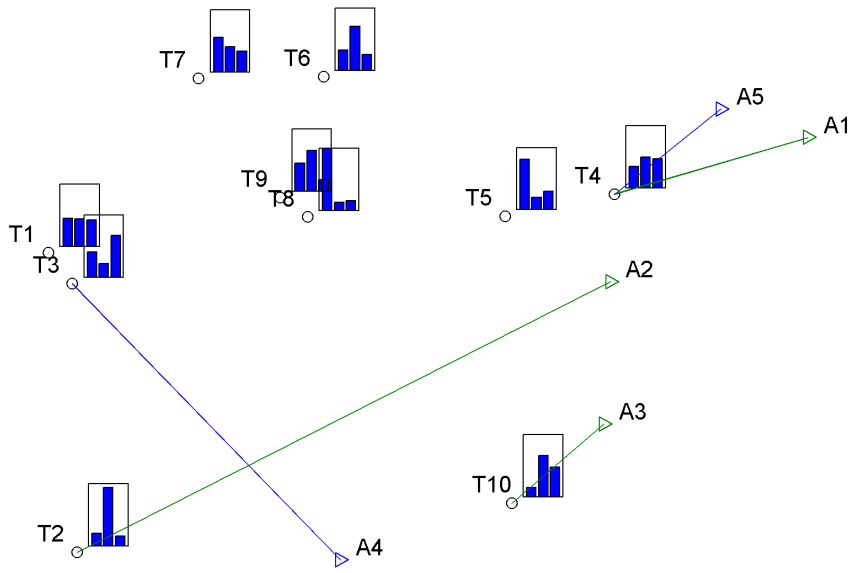


Figure 5-10: Snapshot of decoupled planning result. Green triangles represent tasking agents and blue triangles represent exploration gents. Tasking agents and exploration agents can possibly go to different targets, in which case exploration activities do not fully help to reduce uncertainty of assigned tasks

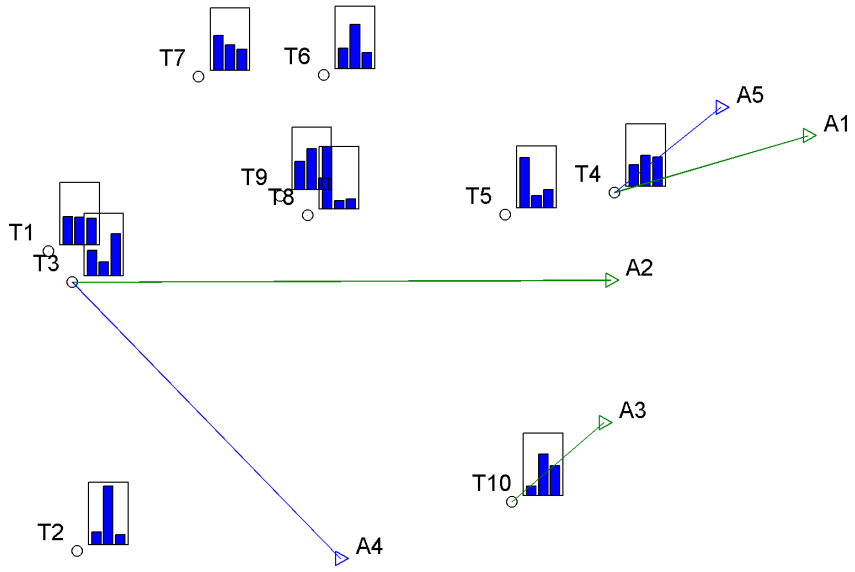


Figure 5-11: Snapshot of coupled planning result. Green triangles represent tasking agents and blue triangles represent exploration gents. Tasking agents group with exploration agents, therefore exploration can always help reduce uncertainty of assigned tasks

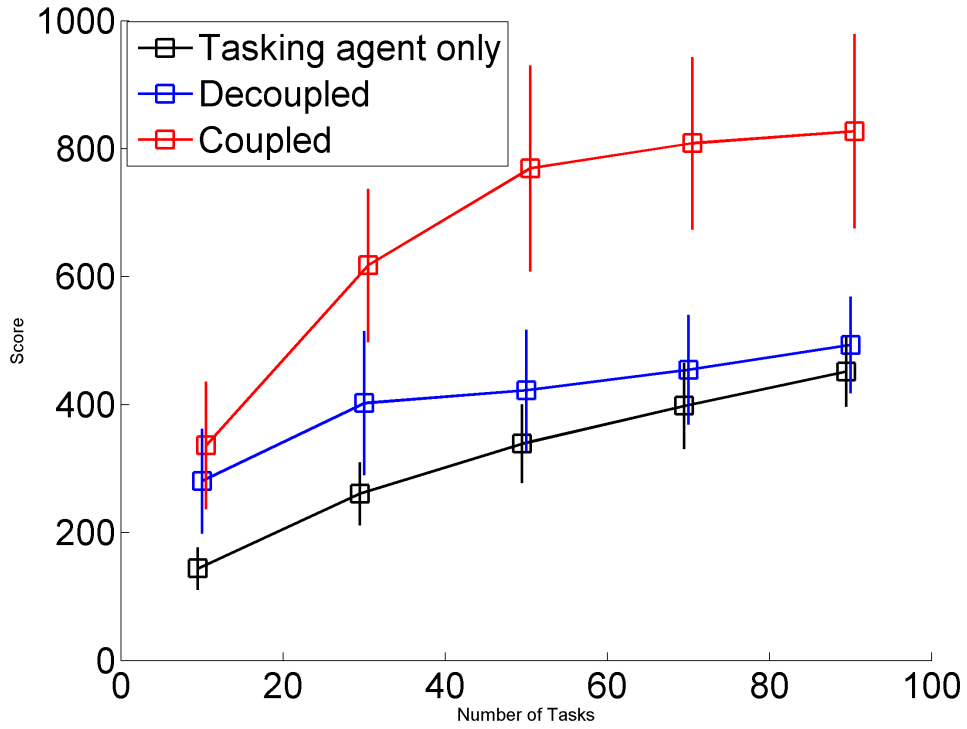


Figure 5-12: Statistics of three planning approaches with different number of targets. Coupled approach gets the highest score. Decoupled approach is in between coupled and tasking agent only approaches. When there are less targets, decoupled approach is closer to coupled approach. When there are more targets, decoupled approach is closer to tasking agent only case.

because it utilizes the heterogeneous abilities of the team and encourages cooperation between them. When there are fewer targets, there is a higher chance that the more rewarding targets are also more uncertain targets, so the exploration vehicles will pair with tasking agents and the performance of decoupled approach is closer to the coupled result. When there are more targets, it is less likely for the exploration agents to pick the same targets as the tasking agents in decoupled approach, and therefore the performance of the decoupled approach is closer to the tasking agent only case.

5.2.3 Hardware Experiments

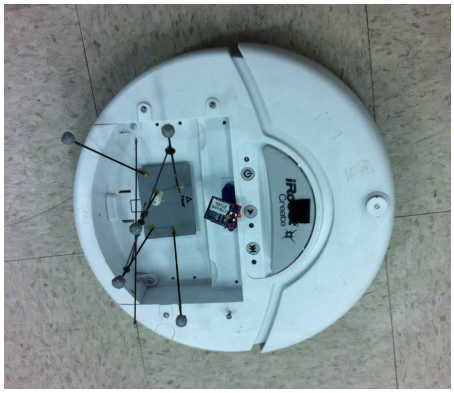
The planning algorithms were also tested using a robot system in the MIT Aerospace Control Laboratory [82]. Figure 5-13 shows an overall view of the hardware system. Figure 5-14a shows the tasking agent. These are iRobot *Create* using bluetooth to communicate with the controller computer. Figure 5-14b shows the exploration agent.

They are P5512 security cameras hanging on the wall of the lab and taking pictures at maximum frequency of 50Hz. These cameras can be tuned to focus on different points of the room by tuning the pan/tilt angles and zoom levels. Moving from one focus point to another takes time and the speed can be specified.

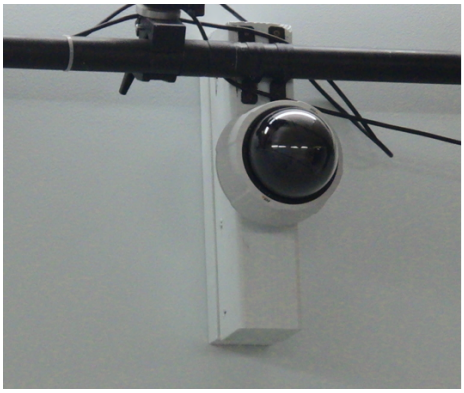
The targets and the measurements model is shown in Figure 5-16. Targets in this testbed are colored papers glued on the floor. Each picture has a dominant color among red, blue, green and yellow indicating the classification of the target. The camera will randomly take a pixel out of the target picture as a measurement of the classification of the target. Since each picture has pixels that are associated with the non-dominant colors, the measurement is not perfect – there is a chance that the camera will fail to identify the classification correctly. Figure 5-15 shows original generated pictures of four different target classes and Figure 5-16a shows an image of a target taken by the camera. In order to filter out background and extract color information, the image is converted from red/green/blue (rgb) format into hue/saturation/value (hsv) format. Background pixels have low saturation. Therefore a saturation threshold is set up to filter the background pixels and leave only target pixels. Figure 5-16b shows the image of the target in Figure 5-16a after converting to hsv format and removing background pixels. Figure 5-16c shows the histogram of hue values of the target pixels in Figure 5-16b. After mapping hue values into four classification colors, Figure 5-16d gives the probability of reporting four classes given the image in Figure 5-16a. It can been seen that the probability of detecting the correct classification γ in Equation (5.6) is about 0.7 and the probability of getting an incorrect classification is about 0.1.

Since the target is not pure in color, the measurement is not perfect (detection rate in Equation (5.6) $\gamma \neq 1$). After taking one measurement and doing Bayesian update (2.10), the posterior probability of wrong classifications is non-zero, and there is still uncertainty in the target. The camera cannot get the right classification immediately after one observation. Therefore, the uncertainty can be further reduced when the camera stays at a target longer and takes more observations. Figure 5-17 shows an example scenario of change in uncertainty of a target (measured by entropy,

Figure 5-13: Hardware testbed in Aerospace Controls Lab., MIT

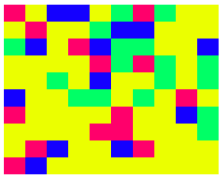


(a) Tasking agent

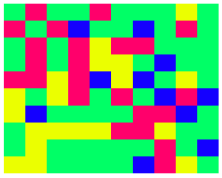


(b) Exploration agent

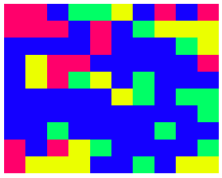
Figure 5-14: hardware components



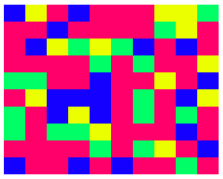
(a) Target class 1



(b) Target class 2

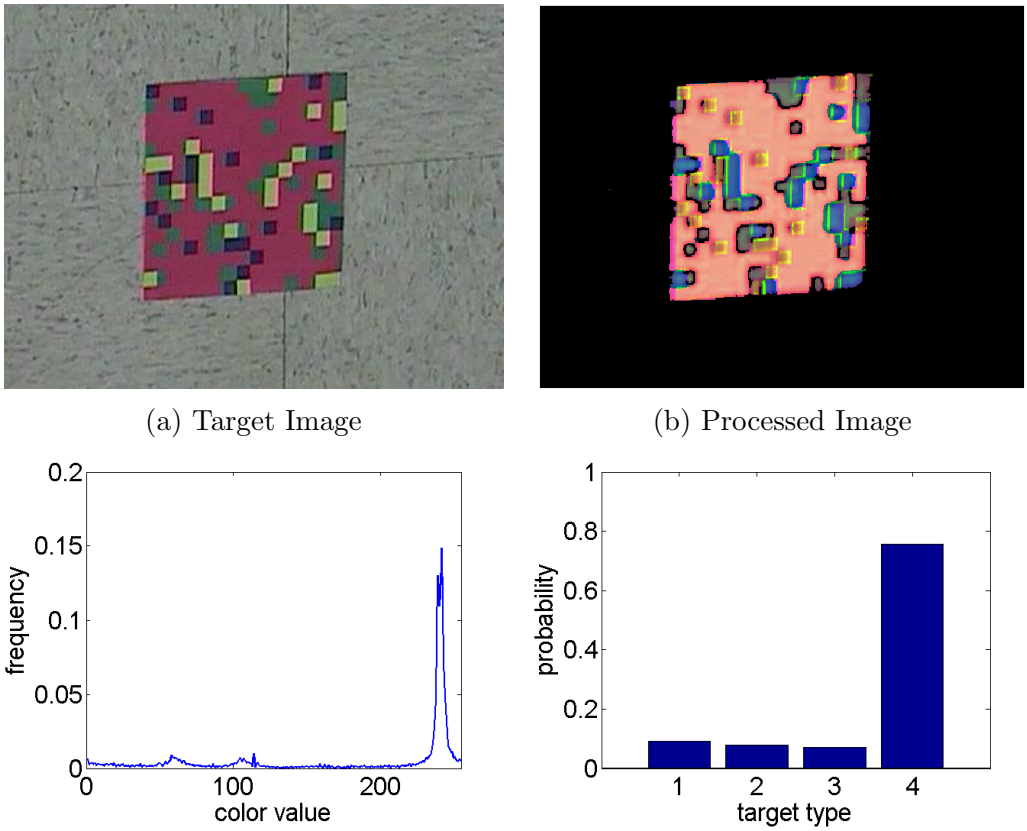


(c) Target class 3



(d) Target class 4

Figure 5-15: Target classification



(a) Target Image (b) Processed Image

(c) Color histogram (d) classification probability

Figure 5-16: Target Measurement Model

see Equation (5.8)) and the number of measurements. In this example, when more measurements are taken, there is less uncertainty about the classification and the score of doing a task over this target is increasing.

In the hardware experiment, there are 2 exploration agents and 2 tasking agents in the system. There are 10 targets, which fall into 4 categories shown in Figure 5-15. The yellow, green, blue and red targets correspond to reward of 5, 20, 30 and 50. The speed of exploration is 1.5 times as fast as that of tasking agent. And the failure chance ϵ defined in Equation (5.5) is set to be 0.05 and the detection rate defined in Equation (5.6) is set to be 0.7.

Table 5.1 shows how decoupled and coupled approaches finish the tasks. It can be seen that they follow different sequences. Decoupled starts with tasks that have higher uncertainty (thus leading to lower scores), and slowly finds the most rewarding targets around 60sec. The mean entropy of the targets of the decoupled approach is

Table 5.1: Planning Algorithm Performance Comparison

Decoupled				Coupled			
Target	Score	Entropy	Time(sec)	Target	Score	Entropy	Time(sec)
T02	5	0.6599	17.1	T08	50	0.1663	13.1
T06	5	0.6599	20.3	T05	50	0.1663	29.9
T07	5	1.0336	30.0	T03	30	0.1663	48.3
T05	50	0.1663	65.4	T10	20	0.0001	70.4
T08	50	0.1663	82.1	T09	20	0.0007	89.0
T10	20	0.1663	88.0	T02	20	0.1225	97.2
T03	30	0.0587	108.4	T06	20	0.0569	110.8
T04	20	0.0357	114.3	T07	5	0.0149	122.4
T01	5	10^{-4}	123.8	T04	20	0.0003	135.9
T09	20	10^{-10}	152.3	T01	5	0.0003	142.8

0.2947. On they other hand, the coupled approach will take observations of potentially more rewarding targets first instead of more uncertain targets, therefore they can find and go to most rewarding targets from the beginning. The mean entropy of targets of coupled approach is 0.0695, significantly less than that of decoupled approach. Figure 5-18 shows the cumulative scores obtained by the two approaches with time. The horizontal axis represents time and the vertical axis represents cumulative score. The VoI based planning algorithm gets higher score in the beginning as it can coordinate agents to most rewarding targets first. The gap between coupled and decoupled approaches narrows down around 60sec as the decoupled approach slowly finds the rewarding targets. Overall the score of coupled approach is always higher than decoupled approach, because agents are more certain about their tasks in the coupled approach.

5.3 Summary

This chapter evaluates algorithms discussed in Chapter 3 and Chapter 4 with both simulated data and real-world data. The numerical results show the new developed sensing and planning algorithms significantly outperform those that are currently available.

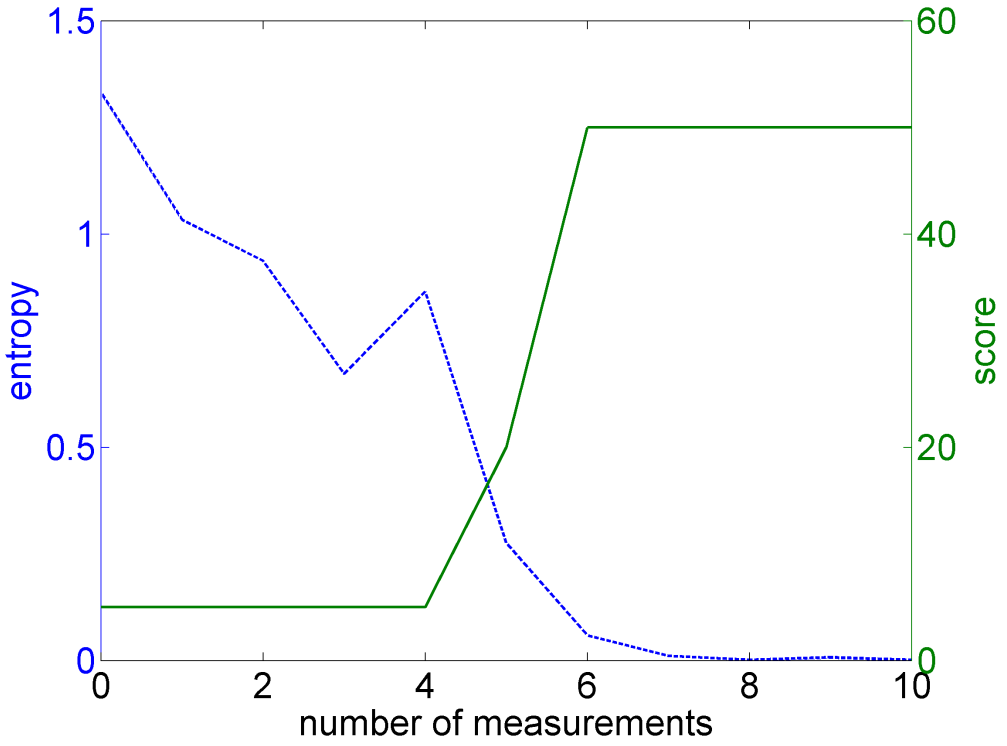


Figure 5-17: Uncertainty decrease and score increases with number of measurements

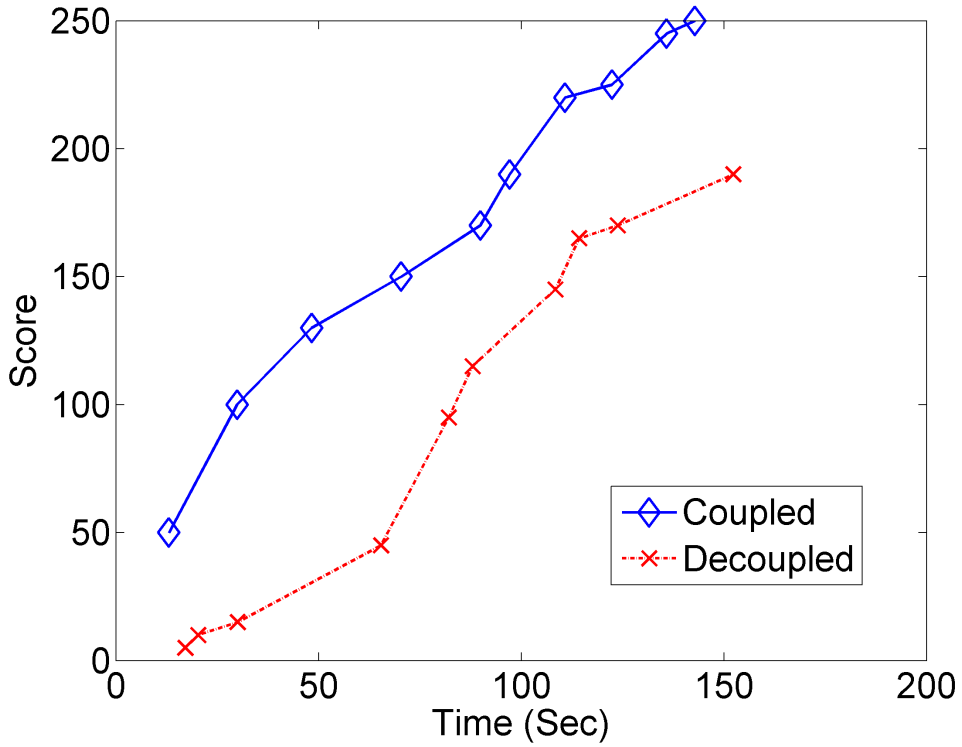


Figure 5-18: Cumulative score of coupled and decoupled approach. Couple approach is able to move towards rewarding targets first, thus outperforms decoupled approach from the beginning. Coupled approach also takes observations before taking actions on targets, therefore has less uncertainty and higher score on average.

Chapter 6

Conclusion and Future Work

This research is motivated by the need to develop more efficient algorithms for performing distributed estimation and planning than those currently available.

A Value of Information based Distributed Sensing (VoIDS) algorithm is presented in which agents identify themselves as *informative* and communicate their information only when the VoI exceeds a threshold. It is proven that the communication cost of VoIDS will asymptotically converge to zero and choice of the VoI threshold gives a upper bound on estimation accuracy. This upper bound drives a dynamic trade-off between the cost of transmitting information, and the accuracy of the final estimate. To accommodate this trade-off, an Adaptive VoI based Distributed Sensing (A-VoIDS) algorithm is developed that adjusts the VoI threshold adaptively to ensure that the available communication bandwidth is better utilized to guarantee asymptotic reduction of estimation error. To further be able to estimate parameters that are changing, Dynamic-VoIDS is developed that continuously forgetting old information to let new information stand out.

Both VoIDS and A-VoIDS are theoretically and experimentally compared with a Full-Relay algorithm, a censoring-based Random Broadcast algorithm, and a Hyperparameter Consensus (HPC) algorithm [15]. Simulation shows that A-VoIDS incurs only a fraction of the communication cost of HPC, while arriving at an even better estimate of the hyperparameters. In case where parameters are changing, it is numerically shown that Dynamic-VoIDS can follow the dynamics in parameters and

get the correct estimate while VoIDS and A-VoIDS cannot. The algorithms are also tested on a real dataset (the Intel temperature dataset [83]), where similar results are obtained. A notable advantage of VoI based algorithms is that they can work on any dynamic network topology, as long as the network remains strongly connected.

An active planning framework is presented for a heterogeneous system with both tasking agents and exploration agents. In particular, a VoI based coupled planning algorithm is developed to assign missions in presence of environment uncertainty. The value of exploration/information gathering is coupled into exploitation of missions, therefore the heterogeneous agents show better cooperation. The framework can well fit into continuous as well as discrete uncertainties, hence goes beyond previous work that deals only with Gaussian uncertainties [35]. A Consensus-Based Bundle Algorithm (CBBA) is used to implement the framework, which can easily incorporate system dynamics such as agent velocities. This VoI based coupled planning algorithm is fully decentralized, and with polynomial computational complexity in the number of targets and agents thus easily scalable.

The coupled algorithm is compared with exploration/tasking decoupled approach by numerical studies. The results show that coupled algorithms can encourage stronger cooperation between heterogeneous agents thus get significant higher score than decoupled approach. These two approaches are further compared in the Aerospace Controls Laboratory hardware testbed, it is shown that coupled algorithm is able to go to high rewarding missions first and use exploration resources to get higher score of these missions.

This work contributes to the goal of developing the next generation intelligent distributed agents. It provides a more efficient framework for performing distributed parameter estimation and mission planning than existing approaches. Furthermore, it is significant to the distributed system literature because it brings in the notion of *censoring* marginally useful information and *coupling* information gathering to mission goals in a distributed framework. The algorithms discussed here, and their possible variants, could lead to significant resource savings and performance promotions in real-world distributed sensing and planning applications by better utilize the

incoming information.

Future work includes:

- Develop performance bounds for Dynamic-VoIDS
- Evaluate developed distributed sensing algorithms in the hardware testbed
- Conduct more experiments to study the effect of parameter settings on the performance of distributed planning algorithms
- Extend planning algorithms to mixed uncertainty types, for example, both the location and classification of a target is uncertain

Bibliography

[1] Brian Mockenhaupt. We've seen the future, and it's unmanned. *Esquire*, September 2009.

[2] Unmanned aircraft systems roadmap: 2007–2032. Technical report, Office of the Secretary of Defense, 2007.

[3] N.E. Leonard, D.A. Paley, F. Lekien, R. Sepulchre, D.M. Fratantoni, and R.E. Davis. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, 2007.

[4] L. Techy, C.A. Woolsey, and D.G. Schmale. Path planning for efficient uav coordination in aerobiological sampling missions. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 2814–2819, 2008.

[5] Nathan Michael, E. Stump, and Kartik Mohta. Persistent surveillance with a team of mavs. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2708–2714, 2011.

[6] X. Wang, V. Yadav, and S. N. Balakrishnan. Cooperative UAV formation flying with obstacle/collision avoidance. *Control Systems Technology, IEEE Transactions on*, 15(4):672–679, 2007.

[7] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte. Decentralized Bayesian negotiation for cooperative search. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[8] A. Makarenko, A. Brooks, T. Kaupp, H. Durrant-Whyte, and F. Dellaert. Decentralised data fusion: A graphical model approach. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 545 –554, july 2009.

[9] R. Murray. Recent research in cooperative control of multi-vehicle systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 2007.

[10] Yongcan Cao, Wei Ren, N. Sorensen, L. Ballard, A. Reiter, and J. Kennedy. Experiments in consensus-based distributed cooperative control of multiple mobile robots. In *International Conference on Mechatronics and Automation*, pages 2819–2824, Aug. 2007.

[11] Reza Olfati-Saber, Alex Fax, and Richard M. Murray. Consensus and cooperation in networked multi-agent systems. *IEEE Transactions on Automatic Control*, 95(1):215–233, January 2007.

[12] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52:2508 – 2530, June 2006.

[13] Magnus Egerstedt and Mehran Mesbahi. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[14] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri. Distributed Kalman filtering using consensus strategies. In *IEEE Conference on Decision and Control (CDC)*, pages 5486–5491, Dec. 2007.

[15] Cameron S.R. Fraser, Luca F. Bertuccelli, Han-Lim Choi, and Jonathan P. How. A hyperparameter consensus method for agreement under uncertainty. *Automatica*, 48(2):374–380, February 2012.

[16] Wei Ren, R. W. Beard, and E. M. Atkins. Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 27(2):71–82, April 2007.

[17] N. Trivedi and N. Balakrishnan. Graphical models for distributed inference in wireless sensor networks. In *Sensor Technologies and Applications, 2009. SENSORCOMM '09. Third International Conference on*, pages 596 –603, june 2009.

[18] S. Grime and H.F. Durrant-Whyte. Data fusion in decentralized sensor networks. *Control Engineering Practice*, 2(5):849 – 863, 1994.

[19] T. Bailey, S. Julier, and G. Agamennoni. On conservative fusion of information with unknown non-gaussian dependence. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 1876 –1883, july 2012.

[20] K. Chang, Chee-Yee Chong, and S. Mori. Analytical and computational evaluation of scalable distributed fusion algorithms. *Aerospace and Electronic Systems, IEEE Transactions on*, 46(4):2022 –2034, oct. 2010.

[21] W.J. Farrell and C. Ganesh. Generalized chernoff fusion approximation for practical distributed data fusion. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 555 –562, july 2009.

[22] S.J. Julier. An empirical study into the use of chernoff information for robust, distributed fusion of gaussian mixture models. In *Information Fusion, 2006 9th International Conference on*, pages 1 –8, july 2006.

[23] Nisar Ahmed, Jonathan Schoenberg, and Mark Campbell. Fast weighted exponential product rules for robust general multi-robot data fusion. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[24] Mark Andrew Paskin. *Exploiting locality in probabilistic inference*. PhD thesis, Berkeley, CA, USA, 2004. AAI3165519.

[25] M. Cetin, L. Chen, J.W. Fisher III, A.T. Ihler, R.L. Moses, M.J. Wainwright, and A.S. Willsky. Distributed fusion in sensor networks. *Signal Processing Magazine, IEEE*, 23(4):42–55, 2006.

[26] Wee Peng Tay, J.N. Tsitsiklis, and M.Z. Win. Asymptotic performance of a censoring sensor network. *Information Theory, IEEE Transactions on*, 53(11):4191 –4209, nov. 2007.

[27] E.J. Msechu and G.B. Giannakis. Distributed measurement censoring for estimation with wireless sensor networks. In *IEEE 12th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 176 –180, June 2011.

[28] Wee Peng Tay. *Decentralized detection in resource-limited sensor network architectures*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2008.

[29] MURAT UNEY. *Decentralized Eestimation Under Ccommunication Constraints*. PhD thesis, MIDDLE EAST TECHNICAL UNIVERSITY, Turkey, 2009.

[30] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden. Distributed regression: an efficient framework for modeling sensor network data. In *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*, pages 1 – 10, april 2004.

[31] C.M. Kreucher, A.O. Hero, K.D. Kastella, and M.R. Morelande. An information-based approach to sensor management in large dynamic networks. *Proceedings of the IEEE*, 95(5):978 –999, may 2007.

[32] Gabriele Oliva, Stefano Panzieri, Attilio Priolo, and Giovanni Ulivi. Adding and removing nodes in consensus. In *Control Automation (MED), 2012 20th Mediterranean Conference on*, pages 1031 –1036, july 2012.

[33] V. Saligrama and M. Alanyali. A token-based approach for distributed computation in sensor networks. *Selected Topics in Signal Processing, IEEE Journal of*, 5(4):817 –832, aug. 2011.

[34] J. Bellingham, M. Tillerson, A. Richards, and J. How. Multi-task allocation and path planning for cooperating UAVs. In *Proceedings of Conference of Cooperative Control and Optimization*, Nov. 2001.

[35] Luca F. Bertuccelli. Robust Planning for Heterogeneous UAVs in Uncertain Environments. Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge MA, June 2004.

[36] M. J. Hirsch, P. M. Pardalos, R. Murphey, and D. Grundel, editors. *Advances in Cooperative Control and Optimization. Proceedings of the 7th International Conference on Cooperative Control and Optimization*, volume 369 of *Lecture Notes in Control and Information Sciences*. Springer, Nov 2007.

[37] J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Enginee. Springer, 2011.

[38] A. Prekopa. *Stochastic Programming*. Kluwer, 1995.

[39] A. Ben-tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25:1–13, 1999.

[40] A. L. Soyster. Convex Programming with Set-Inclusive Constraints and Applications to Inexact Linear Programming. *Operations Research*, 21(5):1154–1157, 1973.

[41] Mehdi Alighanbari. *Robust and Decentralized Task Assignment Algorithms for UAVs*. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge MA, September 2007.

[42] William M. McEneaney and B.G. Fitzpatrick. Control for uav operations under imperfect information. *Proc. 1st AIAA UAV Symposium*, 2002.

[43] Loredana Arienzo. An information-theoretic approach for energy-efficient collaborative tracking in wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.*, 2010:13:1–13:14, January 2010.

[44] Hanbiao Wang, Kung Yao, and Deborah Estrin. Information-theoretic approaches for sensor selection and placement in sensor networks for target localization and tracking. *Communications and Networks, Journal of*, 7(4):438 –449, dec. 2005.

[45] Feng Zhao, Jaewon Shin, and J. Reich. Information-driven dynamic sensor collaboration. *Signal Processing Magazine, IEEE*, 19(2):61 –72, mar 2002.

[46] C. Kreucher, M. Morelande, K. Kastella, and A.O. Hero. Particle filtering for multitarget detection and tracking. In *IEEE Aerospace Conference*, pages 2101 –2116, march 2005.

[47] A. Logothetis, A. Isaksson, and R.J. Evans. An information theoretic approach to observer path design for bearings-only tracking. In *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, volume 4, pages 3132 –3137 vol.4, dec 1997.

[48] Luca Bertuccelli and Jonathan How. Active exploration in robust unmanned vehicle task assignment. *Journal of Aerospace Computing, Information, and Communication*, 8:250–268, 2011.

[49] Beipeng Mu, Girish Chowdhary, and Jonathan P. How. Efficient distributed information fusion using adaptive decentralized censoring algorithms. In *American Control Conference (ACC)*, Washington DC, July 2013 (accepted). IEEE.

[50] Beipeng Mu, Girish Chowdhary, and Jonathan P. How. Efficient distributed inference using adaptive censoring algorithms. *Automatica*, 2013 (submitted).

[51] Beipeng Mu, Girish Chowdhary, and Jonathan P. How. Value-of-information aware active task assignment. In *AIAA Guidance Navigation and Control*, Boston, August 2013 (to appear).

[52] Beipeng Mu, Girish Chowdhary, and Jonathan P. How. Value-of-information aware active task assignment. In *Defense Security and Sensing*, Baltimore, May 2013. SPIE.

[53] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis.* Chapman and Hall, 1995.

[54] Lorraine Schwartz. On bayes procedures. *Probability Theory and Related Fields*, 4:10–26, 1965. 10.1007/BF00535479.

[55] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, January 2008.

[56] L. D. Brown. *Fundamentals of statistical exponential families: with applications in statistical decision theory.* Institute of Mathematical Statistics, Hayworth, CA, USA, 1986.

[57] Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication.* The University of Illinois Press, Urbana, IL, 1948.

[58] Solomon Kullback. *Information Theory and Statistics.* John Wiley and Sons, New York, 1959.

[59] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[60] A. Renyi. On measures of information and entropy. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, pages 547–561, 1960.

[61] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesisbased on the sum of observations. *Annals of Mathatical Statistics*, 23(4):493–507, 1952.

[62] S. M. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, 28(1):131–142, 1966.

[63] A.O. Hero III, C.M. Kreucher, and D. Blatt. Information theoretic approaches to sensor management. *Foundations and Applications of Sensor Management(Signals and Communication Technology Series). Berlin, Springer Science+ Business Media Deutschland GmbH, 2007.*, pages 33–57, 2007.

[64] A. Hero, D. Castanon, D. Cochran, and K. Kastella. *Foundations and Applications of Sensor Management.* Springer-Verlag, 2008.

[65] S. Martínez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4):661–668, April 2006.

[66] Allison D. Ryan, Hugh F. Durrant-Whyte, and J.K. Hedrick. Information-theoretic sensor motion control for distributed estimation. In *Proc. ASME Int. Mechanical Engineering Congress and Exposition*, pages 725–734, 2007.

[67] F. Nielsen and R. Nock. Entropies and cross-entropies of exponential families. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 3621 –3624, sept. 2010.

[68] Vijay Gupta. *Distributed Estimation and Control in Networked Systems*. PhD thesis, California Institute of Technology, 2006.

[69] A. Muhammad and A. Jadbabaie. Decentralized computation of homology groups in networks by gossip. In *American Control Conference*, 2007.

[70] Marzieh Nabi Abdolyousefi and Mehran Mesbahi. Network identification via node knockout. In *IEEE Conference on Decision and Control*, 2010.

[71] Girish Chowdhary, Magnus Egerstedt, and Eric N. Johnson. Network discovery: An estimation based approach. In *American Control Conference*, San Francisco, CA, June 2011.

[72] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. Wiley, TK5102.5.B696. 1992.

[73] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, second edition, 2003.

[74] Pavlo Krokhmal, Robert Murphey, Panos Pardalos, Stanislav Uryasev, and Grigory Zrazhevski. Robust decision making: Addressing uncertainties. In *in Distributions", In: S. Butenko et al. (Eds.) "Cooperative Control: Models, Applications and Algorithms*, pages 165–185. Kluwer Academic Publishers, 2003.

[75] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2007.

[76] E. Delage and S. Mannor. Percentile optimization for markov decision processes with parameter uncertainty. *Operations research*, 58(1):203–213, 2010.

[77] L. Blackmore and M. Ono. Convex chance constrained predictive control without sampling. *AIAA Proceedings.[np]. 10-13 Aug*, 2009.

[78] Sameera S. Ponda, Luke B. Johnson, and Jonathan P. How. Distributed chance-constrained task allocation for autonomous multi-agent teams. In *American Control Conference (ACC)*, June 2012.

[79] H.-L. Choi, L. Brunet, and J. P. How. Consensus-based decentralized auctions for robust task allocation. *IEEE Transactions on Robotics*, 25(4):912–926, August 2009.

[80] Han-Lim Choi. *Adaptive Sampling and Forecasting With Mobile Sensor Networks*. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge MA, February 2009.

[81] Peter Bodik, Wei Hong, Carlos Guestrin, Sam Madden, Mark Paskin, and Romain Thibaux. Intel lab data. Technical report, February 2004.

[82] J. P. How, B. Bethke, A. Frank, D. Dale, and J. Vian. Real-time indoor autonomous vehicle test environment. *IEEE Control Systems Magazine*, 28(2):51–64, April 2008.

[83] Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-optimal sensor placements in gaussian processes. In *International Conference on Machine Learning (ICML)*, 2005.