Decentralized Multi-Agent Active Search for Sparse Signals

Ramina Ghods[1] **Arundhati Banerjee**[1] **Jeff Schneider**[1]

[1]School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA

Abstract

Active search refers to the problem of efficiently locating targets in an unknown environment by actively making data-collection decisions. In this paper, we are focusing on multiple aerial robots (agents) detecting targets such as gas leaks, radiation sources or human survivors of disasters. One of the main challenges of active search with multiple agents in unknown environments is impracticality of central coordination due to the difficulties of connectivity maintenance. In this paper, we propose two distinct active search algorithms that allow for multiple robots to independently make data-collection decisions without a central coordinator. Throughout we consider that targets are sparsely located around the environment in keeping with compressive sensing assumptions and its applicability in real world scenarios. Additionally, while most common sensing algorithms assume that agents can sense the entire environment (e.g. compressive sensing) or sense point-wise (e.g. Bayesian Optimization) at a time, we make a realistic assumption that each agent can only sense a contiguous region of space at each time step. We provide simulation results as well as theoretical analysis to demonstrate the efficacy of our proposed algorithms.

1 INTRODUCTION

Active search (active sensing) defines the problem of efficiently locating targets in an unknown environment by interactively collecting data and finds use in applications such as detecting gas leaks, pollution sources or search and rescue missions [Rolf et al., 2018, Flaspohler et al., 2019, Ma et al., 2017]. Originally, much of the work in the field of robotics information gathering had their focus on single-agent settings [Cliff et al., 2015, Lim et al., 2016, Patten et al., 2018, Arora et al., 2019], or if they were multi-agent, they required a central planner [Cho et al., 2018, Charrow et al., 2014, Surmann et al., 2019]. While centralized planning is one approach to multi-robot settings, it is often impractical in certain applications of surveillance, exploration of unknown environments and search-and-rescue [Sabattini et al., 2013]. This is because in these applications connectivity maintenance is specifically difficult [Yan et al., 2013, Robin and Lacroix, 2016]. Moving in an unknown or cluttered environment, it is very likely for robots to get trapped and temporarily lose their connection to the center [Sabattini et al., 2013]. As a result, a central coordinator that expects synchronicity from all robots at all times is not feasible as any agent failure or communication delay could disrupt the entire process [Queralta et al., 2020, Best et al., 2019, Lauri et al., 2020, Murphy, 2004, Feddema et al., 2002]. To clarify, there is still communication between agents to share information, otherwise they are just independent actors, not a team. In this paper, we propose two multi-robot active search algorithms that allow agents to make independent and intelligent decisions in a decentralized manner.

Another consideration of this paper is a realistic assumption on the sensing actions called region sensing. Inspired by aerial robots' field of view, we assume that each agent senses an average value of a contiguous region (block) of the space at each time step [Ma et al., 2017]. The size of the sensing block models the distance of the agent from the region. We also model noise in the observations in accordance to this distance. Specifically, we assume sensing a larger contiguous region (modeling farther distance from region) inflicts a larger noise value on the resulting observation. Lastly, as an essential part of the real-world applications of active search, we assume targets are sparsely located around.

Contributions

- We propose two novel algorithms, SPATS (Sparse Parallel Asynchronous Thompson Sampling) and LATSI (LAplace Thompson Sampling with Information gain), to actively locate targets in an unknown environment.

Accepted for the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021).

SPATS is an online algorithm with a probabilistic exploration approach that does not need any prior information about the signal of interest. LATSI leverages the benefits of mutual information with probabilistic exploration in the search space.

- SPATS and LATSI have 3 main features that collectively distinguish them in robotics applications of active search. 1) They are multi-agent methods with agents asynchronously making independent data-collection decisions without a central planner. 2) They are developed with a practical region sensing assumption. 3) They consider sparse signal recovery.
- We demonstrate the efficacy of SPATS and LATSI with an extensive set of simulation results in an asynchronous multi-agent setting. We provide theoretical analysis on the benefits of SPATS.

While there have been many sparse recovery algorithms proposed in the literature, to the best of our knowledge there is no algorithm proposed that develops sparse estimators for active learning methods with multi-agent structure and region sensing assumptions. In this paper, we show how sparsity in its nature limits the exploration factor in active learning methods and how a practical region sensing assumption exacerbates this situation. We propose SPATS and LATSI to strategically address such problems to successfully recover sparse signals.

1.1 RELATED WORK

A prominent approach to estimating sparse signals is compressive sensing (CS) [Candès et al., 2006, Donoho, 2006]. There has been a large number of work on adaptive CS that enables the ability to make online and adaptive measurements to estimate sparse signals and thus is applicable to active search problems [Braun et al., 2015, Haupt et al., 2009a,b, Davenport and Arias-Castro, 2012, Malloy and Nowak, 2014]. Unfortunately, such adaptive CS methods are sequential and therefore not extendable to multi-agent scenarios. Furthermore, CS algorithms in general assume that every measurement can sense the entire environment with arbitrary coefficients which is not a practical assumption for active search problems with region sensing constraints.

Another area of work are multi-armed bandits. Abbasi-Yadkori et al. [2012] and Carpentier and Munos [2012] provide multi-armed bandit algorithms that include a sparsity assumption on their hyperparameter. However, the focus of these algorithms are not on estimating the sparse parameter. There have been other Bayesian Optimization (BO) and active learning methods proposed for active search. Marchant and Ramos [2012] uses BO to develop a spatial mapping of a region whereas we are interested in locating targeted signals. Carpin et al. [2015] uses BO for localization of single wireless devices but only focuses on point sensing actions. Ma et al. [2017], Rajan et al. [2015], Jedynak et al. [2012] aim at locating targets by optimizing some notion of Shannon information. Unfortunately, all of the aforementioned active learning algorithms are developed for single agent applications, and except for Ma et al. [2017], they mostly lack any realistic assumptions on sensing actions.

In multi-agent active learning, algorithms in general require a central planner to optimize a batch of actions for all agents at each time step and therefore are not applicable to our problem setting [Azimi et al., 2012, Gu et al., 2014, Azimi et al., 2010]. Another multi-agent area of work is mobile sensor networks (MSN) [Nguyen, 2019, Chen et al., 2019, La et al., 2014] where multiple mobile sensors/agents reconstruct a scalar map of sensory values in an entire area. MSNs typically consider some form of region sensing assumption on their actions, however, they generally have a constricting sensor network with strict communication patterns which differentiates them from our applications.

In robotics, methods that deal with active search generally aim at autonomously building topological (identify obstacles and clearways) and/or spatial maps of a region. Our active search differs from topological mapping techniques such as SLAM [Leonard and Durrant-Whyte, 1991, Huang et al., 2019] and can be most closely related to spatial mapping. For example, Rolf et al. [2018] identify strong signals in environments with background information using trajectory planning with confidence intervals; but, unlike our setting, their algorithm is developed for a single agent performing point sensing observations. In the area of robotics information gathering, there has been more attention towards the need for decentralized solutions recently [Queralta et al., 2020, Zhang et al., 2019]. However, existing methods for decentralized multi-agent systems either assume reliable communication requirements to share future plans [Best et al., 2019, Dames et al., 2017, Li and Duan, 2017] or assume centralized sharing of observations following decentralized execution [Lauri et al., 2020, Lowe et al., 2017]. Our algorithms benefit from observation sharing when it occurs, but never depend on communication for coordination.

Notation Lowercase and uppercase boldface letters represent column vectors and matrices, respectively. For a matrix $\mathbf{A}$, its transpose is $\mathbf{A}^{\mathrm{T}}$. The ℓ_1 and ℓ_2-norm of vector $\mathbf{a}$ are denoted by $\|\mathbf{a}\|_1$ and $\|\mathbf{a}\|_2$. The $N \times N$ identity matrix is denoted by $\mathbf{I}_N$. The Kronecker product is $\otimes$, and $\mathrm{diag}(\mathbf{a})$ is a square matrix with $\mathbf{a}$ on the main diagonal. For a set $\mathcal{S}$, $|\mathcal{S}|$ denotes number of elements in that set.

2 PROBLEM FORMULATION

Figure 1 illustrates a multi-agent active search problem for a two-dimensional environment. Our goal is to efficiently search for targets in an unknown environment by actively taking sensing actions given all the observations thus far.

This can be thought of as an *active learning* problem (referred to as "Design of Experiment" in statistical literature)[Settles, 2009]. In particular, we are interested in recovering the sparse d-dimensional matrix $\mathbf{S} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ representing the unknown environment. We have no knowledge on the true prior distribution of matrix $\mathbf{S}$ other than knowing it is sparse. Defining $\boldsymbol{\beta} \in \mathbb{R}^n$ as a flattened (vectorized) version of matrix $\mathbf{S}$ with $n = n_1 \times \ldots \times n_d$, we can write each sensing operation at time step t as:

$$y_t = \mathbf{x}_t^{\mathrm{T}} \boldsymbol{\beta} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2), t = 1, ..., T. \tag{1}$$

Here, y_t is the observation and vector $\mathbf{x}_t \in \mathbb{R}^n$ is the sensing action at time step t. We call the set of $(\mathbf{x}_t, y_t)$ the measurement at time step t. Our objective is to estimate the k-sparse vector $\boldsymbol{\beta}$ ($k \ll n$) with as few number of measurements T as possible. Here, we are interested in rectangular sensing actions referred to as *region sensing* [Ma et al., 2017]. Precisely, in the original d-dimensional space, our sensing action will be a d-dimensional contiguous rectangle (region) with weights w_t inside the rectangle and zeros outside. As an example, if $d = 1$, the sensing action becomes $\mathbf{x}_t = [0, ..., 0, w_t, ..., w_t, 0, ..., 0]^{\mathrm{T}}$. This constraint models a robot sensing a region of the search space as illustrated in Figure 1. Furthermore, we dedicate a fixed amount of power to each sensing action by letting $\|\mathbf{x}_t\|_2 = 1$ (see Remark 1).

Remark 1. *By dedicating a fixed amount of power to each sensing action, we are modeling noise as a function of distance from the region. In particular, by standing at a farther distance from the area, an agent can cover a larger region in an observation. Spreading the fixed amount of sensing power over this larger region would result in larger noise on the observation (a lower resolution observation). Similarly, sensing a smaller region at a closer distance to the environment would model smaller noise (higher resolution observation). Figure 1 illustrates practicality of this model.*

2.1 MULTI-AGENT SETTING

Communication Setup In order to achieve the objective above with multiple agents, we need to first describe our communication setup which is motivated by real outdoor multi-aerial robot systems in field tests. Despite unreliability in unknown environments, communication becomes available sometimes and we want to take advantage of it when possible. That leads to the following constraints for our algorithm: 1) Agents share their past actions and observations when possible. 2) There can be no requirement that the set of available past measurements remains consistent across agents since communication problems can prevent it. 3) There can be no part of the algorithm where an agent must wait for communication from its teammates before acting since this wait could be arbitrarily long and thus cause a loss of valuable sensing time.

We are now ready to describe the multi-agent setting. To actively locate targets, at each time step $t \leq T$, we choose a sensing action $\mathbf{x}_t$ given all the available measurements thus far. For a single agent this procedure is sequential as in Figure 2a where at time step t the agent uses all previous sequential measurements $\mathbf{D}_t^1 = \{(\mathbf{x}_{t'}, y_{t'}) \,|\, t' = \{1, .., t-1\}\}$ to make a decision. The superscript in $\mathbf{D}_t^1$ indicates the agent index. In this paper, however, we are interested in an asynchronous parallel approach with multiple agents independently making data-collection decisions as in Figure 2b. Here, asynchronicity means that agents don't wait on results from other agents; instead, an agent starts a new query immediately after its previous data acquisition is completed using all the measurements available thus far; e.g. in Figure 2b, second agent queries $t = 6$'th action before tasks 4 and 5 are completed using available measurements $\mathbf{D}_6^2 = \{(\mathbf{x}_{t'}, y_{t'}) \,|\, t' = \{1, 2, 3\}\}$.

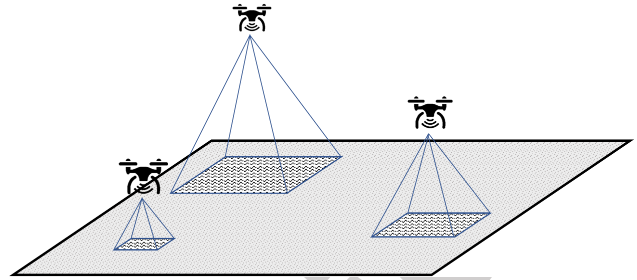


Figure 1: An illustration of multi-agent active search. Multiple aerial robots are sensing an area looking for targets. Agents are free to move in all directions. If an agent moves farther from the region, it can cover a larger portion in one lower-resolution observation. Moving closer to the region covers a smaller region in one higher-resolution observation.

For easier computations, we can write a compact model of all the available measurements in $\mathbf{D}_t^j$ for agent j. For example for sequential $\mathbf{D}_t^1$, by defining $\mathbf{y} = [y_1, ..., y_{t-1}]^{\mathrm{T}}$, $\mathbf{X} = [\mathbf{x}_1^T, ..., \mathbf{x}_{t-1}^{\mathrm{T}}]^{\mathrm{T}}$ we can write the model in (1) as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_{t-1}). \tag{2}$$

3 THOMPSON SAMPLING FOR ACTIVE SEARCH

Thompson Sampling (TS) is an exploration-exploitation algorithm originally introduced for clinical trials by Thompson [1933] and later rediscovered for multi-armed bandits [Wyatt, 1998, Strens, 2000, Russo et al., 2018]. The idea of TS is to balance between exploration and exploitation by maximizing the expected reward of its next action assuming that a sample from the posterior is the true state of the world [Russo et al., 2018]. This feature makes TS an excellent candidate for our asynchronous multi-agent setup. Essentially, by using TS's posterior samples in our reward function, we enable a calculated randomness in each agent's reward function. As a result, multiple agents can take independent samples and therefore solve for different reward

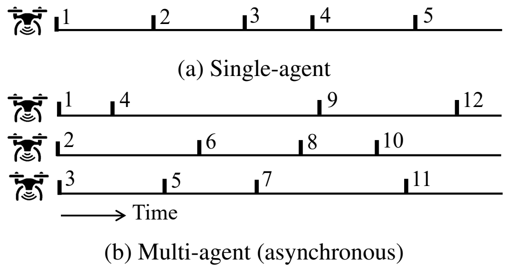


Figure 2: Single vs. multi-agent – Here, the small vertical lines indicate the start of t'th task. In single agent, tasks start sequentially. In multi-agent, task t can start before all previous $t-1$ tasks are finished.

values that equally contribute to the overall goal. Kandasamy et al. [2018] has used this feature of TS for BO to develop an asynchronous yet centralized parallel setting. We instead propose to develop TS in a decentralized and asynchronous setting where each agent independently makes decisions given the measurements available to it, i.e. $\mathbf{D}_t^j$. We will develop our algorithm in the presence of sparsity and region sensing considerations which are troublesome assumptions as will be apparent in the next subsection.

Because TS was originally proposed for bandit problems, one might question its ability on active search and assume it might keep exploiting the same target. However, we are here interested in an adaptation of TS to parameter learning which is in fact perfect for active search. Having attracted a lot of attention in the past decade, TS has been successfully adapted to a variety of online learning problems [Russo et al., 2018]. Our active search problem falls in the category of parameter estimation in active learning as developed by Kandasamy et al. [2019] with the name Myopic Posterior Sampling (MPS). Similar to MPS, our goal is to actively learn (estimate) parameter $\boldsymbol{\beta}$ by taking as few measurements as possible. Since the goal of MPS is to learn parameter $\boldsymbol{\beta}$, its reward function is designed to keep exploring the space as long as there are unexplored (or loosely explored) locations in the parameter space (will not get stuck exploiting). We will next derive MPS for an asynchronous multi-agent setting. For the sake of similarity, we will use TS to refer to MPS.

Algorithm 1 Asynchronous Multi-Agent TS

Assume: prior $\boldsymbol{\beta} \sim p_0$ and likelihood $p(y_t|\mathbf{x}_t, \boldsymbol{\beta})$
For $t = 1, ..., T$
  Wait for an agent to finish; **For** the free agent j:
    Sample $\boldsymbol{\beta}^\star \sim p(\boldsymbol{\beta}|\mathbf{D}_t^j)$ *(Posterior Sampling)*
    Select $\mathbf{x}_t = \arg\max_{\mathbf{x}} \lambda^+(\boldsymbol{\beta}^\star, \mathbf{D}_t^j, \mathbf{x})$ *(Design)*
    Observe y_t given action $\mathbf{x}_t$
    Update & share measurements $\mathbf{D}_{t+1}^j = \mathbf{D}_t^j \cup (\mathbf{x}_t, y_t)$

3.1 DECENTRALIZED MULTI-AGENT TS

We now review TS for active learning and then develop it to a decentralized asynchronous (async.) multi-agent setting. We start with the single agent setting as introduced by Kandasamy et al. [2019]. We are interested in recovering the n-dimensional vector $\boldsymbol{\beta} \sim p_0$. We actively query actions $\mathbf{x}_t$ and observe their outcome y_t where the likelihood $p(y_t|\mathbf{x}_t, \boldsymbol{\beta})$ is known. To query the best action, we maximize a reward function $\lambda(\boldsymbol{\beta}^\star, \mathbf{D}_t^1)$. As an example, the reward function can be $\lambda(\boldsymbol{\beta}^\star, \mathbf{D}_t^1) = -\|\boldsymbol{\beta}^\star - \hat{\boldsymbol{\beta}}(\mathbf{D}_t^1)\|_2^2$, where $\boldsymbol{\beta}^\star$ is our belief of the true $\boldsymbol{\beta}$, and $\hat{\boldsymbol{\beta}}(\mathbf{D}_t^1)$ is the estimated value of $\boldsymbol{\beta}$ given all the available measurements $\mathbf{D}_t^1$ (e.g. maximum likelihood estimate). We are interested in the myopic policy which selects action $\mathbf{x}$ that maximizes the expected reward of time step t, i.e.

$$\lambda^+(\boldsymbol{\beta}^\star, \mathbf{D}_t^1, \mathbf{x}) = \mathbb{E}_{y|\mathbf{x},\boldsymbol{\beta}^\star}\left[\lambda(\boldsymbol{\beta}^\star, \mathbf{D}_t^1 \cup (\mathbf{x}, y))\right]. \quad (3)$$

Here, the best reward would be the one that has access to the true value of $\boldsymbol{\beta}$, i.e. $\lambda^+(\boldsymbol{\beta}, \mathbf{D}_t^1, \mathbf{x})$. Not knowing the true value of $\boldsymbol{\beta}$, TS will sample it from the current posterior distribution of $\boldsymbol{\beta}$ conditioned on the measurements $\mathbf{D}_t^1$, i.e. $\boldsymbol{\beta}^\star \sim p(\boldsymbol{\beta}|\mathbf{D}_t^1)$. Then, TS will pick the sensing action $\mathbf{x}_t$ that maximizes the reward (3) using the sample $\boldsymbol{\beta}^\star$ as its belief of true $\boldsymbol{\beta}$. For the case of multi-agent, consider J agents planning on taking T measurements of an environment. Say agent j finishes making an observation and is ready to choose the t'th action. Using measurements available to this agent so far ($|\mathbf{D}_t^j| \leq t-1$), it will update and sample the posterior (*posterior sampling*), select its next sensing action that maximizes the reward (*design*), evaluate its action and share the observations with other agents. Algorithm 1 summarizes this process.

3.2 THOMPSON SAMPLING WITH SPARSITY

To perform search and rescue, traditionally people have used coverage planning methods with exhaustive search [Lin and Goodrich, 2009, Chien et al., 2010, Ryan and Hedrick, 2005]. However, with the availability of high and low resolution observation points, an optimized active search method can locate targets faster than exhaustive search in terms of number of observations (See Section 5 and Figure 3b). Such faster recovery is achievable due to the concept of sparse signal recovery (compressive sensing) which says that we can recover a sparse signal with length n by taking less than n low or high resolution measurements [Candès et al., 2006, Donoho, 2006]. By using sparsity as the prior information for TS, we can create the right balance between exploring larger regions with low resolution and then exploiting the ones we suspect of including a target with a closer look (higher resolution observation). We will next develop TS in Algorithm 1 for our active search problem in Section 2 using a sparse prior.

We start by first establishing the prior $p(\boldsymbol{\beta})$ and likelihood distribution $p(y_t|\mathbf{x}_t, \boldsymbol{\beta})$. As for the prior, our knowledge is limited to the presence of sparsity. Hence, we will assume $\boldsymbol{\beta}$ has a Laplace distribution with independent entries and a tunable parameter b, i.e. $p(\boldsymbol{\beta}) = \frac{1}{(2b)^n} \exp(-\frac{\|\boldsymbol{\beta}\|_1}{b})$. Laplace distribution translates to an ℓ_1-norm regularization term in the cost function which has been shown to introduce sparsity into the estimator [Williams, 1995, Tibshirani, 1996, Chen et al., 2001]. For the likelihood distribution, the sensing model in (2) gives $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}) = \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_{t-1})$. Next step is to derive the posterior sampling and design stages of Algorithm 1 using this prior and likelihood. Appendix 1 provides a detailed derivation of these two stages. We call the resulting algorithm Laplace-TS.

Facing the Failure Mode of TS with Single Agent Unfortunately, Laplace-TS with single agent leads to poor performance that is on par with a point-wise algorithm that exhaustively searches all locations one at a time. We can associate this poor performance with one of the failure modes of TS discussed in Sec. 8.2 of the tutorial by Russo et al. [2018]. According to the tutorial, TS faces a dilemma when solving certain kinds of active learning problems. One such scenario are problems that require a careful assessment of information gain. In general, by optimizing the expected reward, TS always restricts its actions to those that have a chance in being optimal which in our case are sparse sensing actions restricted further by the region sensing constraint. However, in active learning problems such as ours, suboptimal actions (i.e. nonsparse sensing actions) can carry additional information regarding the parameter of interest. Appendix 2 includes simulation results as well as an example to further illustrate the failure mode in active search. In the next section, we will modify Laplace-TS and propose two algorithms that can bypass this failure mode.

4 OUR PROPOSED ALGORITHMS

4.1 SPATS: SPARSE PARALLEL ASYNC. TS

Per our discussion in Section 3.2, introducing sparsity into TS algorithm limited its ability to explore queries. With this in mind, one might conclude that choosing non-sparse samples in the posterior sampling stage of Algorithm 1 should solve this problem. However, this strategy will still face the failure mode of TS because it is the sparse estimator in the design stage that is limiting the feasible sensing actions. The next logical solution would then be to make both the estimator and posterior sampling procedures non-sparse. Even though with this strategy we will avoid the failure mode of TS, without taking advantage of the prior information about sparsity, the resulting non-sparse TS will be performing no better than exhaustively searching the entire space. To overcome this issue, we propose making an assumption on the prior distribution of both the sampling and estimation procedures that the neighbouring entries of the sparse vector $\boldsymbol{\beta}$ are spatially correlated, i.e. $\boldsymbol{\beta}$ is block sparse. Such spatial correlation creates the most compatible results to the region sensing constraint which only approves sensing actions with a single non-zero block of sensors. Furthermore, we expect block sparsity to introduce exploration ability while also keeping sparsity a useful information in the recovery process. In particular, by gradually reducing the length of the blocks from a large value, we gently trade exploration with exploitation capability over time.

In short, borrowing ideas from a block sparse Bayesian framework introduced by Zhang and Rao [2011], we use a block sparse prior $p(\boldsymbol{\beta}) = \mathcal{N}(\mathbf{0}_{n\times 1}, \boldsymbol{\Sigma}_0)$, where: $\boldsymbol{\Sigma}_0 = \mathrm{diag}\left([\gamma_1 \mathbf{B}_1, ..., \gamma_M \mathbf{B}_M]\right)$, with γ_m and $\mathbf{B}_m \in \mathbb{R}^{L\times L}$ $(m = 1, ..., M)$ as hyperparameters. Here, γ_m controls the sparsity of each block as is the case in sparse Bayesian learning methods [Tipping, 2001, Wipf and Rao, 2004], i.e. when $\gamma_m = 0$, the corresponding block m is zero. Here, L is the length of the blocks that we will gradually reduce in the TS process. To avoid overfitting while estimating these hyperparameters, Zhang and Rao [2013] suggests one matrix $\mathbf{B}$ to model all block covariances, namely $\boldsymbol{\Sigma}_0 = \mathrm{diag}(\boldsymbol{\gamma}) \otimes \mathbf{B}$, where, $\boldsymbol{\gamma}$ is the vector containing all elements of γ_m for $m = 1, ..., M$. Appendix 3 provides a detailed derivation of Algorithm 1 with this prior for the active search problem in Section 2. Algorithm 2 called SPATS summarizes our results in this section. SPATS has much lower computational cost than Laplace-TS algorithm since it does not require a Gibbs sampler. Furthermore, unlike Laplace-TS, SPATS does not need to know the sparsity rate or any other prior information about the true signal $\boldsymbol{\beta}$.

4.1.1 Theoretical Analysis of a Sparse Model

We now provide theoretical analysis testifying to the benefits of SPATS. SPATS has two aspects that distinguish it from a naïve TS developed for sparse signals. One is using a block sparse prior with varying block length and two is using multiple agents. In what follows, we introduce two theorems to investigate the benefits of each aspect separately. First in Theorem 1, for a sparse model with single agent setting we will compute and compare upper bounds on the expected regret of two TS algorithms with a 1-sparse and a 1-block sparse prior with one nonzero block. The 1-block sparse prior closely imitates SPATS's performance with a region sensing assumption. See proof in Appendix 4.

Theorem 1. *Consider an active search problem with a 1-sparse true parameter $\boldsymbol{\beta} \in \mathbb{R}^n$ and reward function $\mathcal{R}(\mathbf{x}, \boldsymbol{\beta}) = (\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})^2$ for action $\mathbf{x} \in \mathbb{R}^n$ chosen from set of actions $\mathcal{X}$ that satisfy region sensing in Section 2. Consider two single agent TS algorithms where one assumes a 1-sparse prior and another uses a 1-block sparse prior with varying block length as defined in Algorithm 2. Then, the*

Algorithm 2 SPATS

Assume: Sensing model (1); sparse signal β; J agents; block length L
Set: $\mathbf{D}_0^j = \varnothing$; $L = n/J$; $\gamma_m = 1$; $\mathbf{B}$: random highly correlated covariance matrix
For $t = 1, ..., T$
  Wait for an agent to finish; **For** the free agent j:[1]
    Sample $\boldsymbol{\beta}^\star \sim p(\boldsymbol{\beta}|\mathbf{D}_t^j, \boldsymbol{\gamma}, \mathbf{B})$ from equation (14)
    Select $\mathbf{x}_t = \arg\max_{\mathbf{x}} \lambda^+(\boldsymbol{\beta}^\star, \mathbf{D}_t^j, \mathbf{x})$ using (16)
    Observe y_t given action $\mathbf{x}_t$
    Update & share measurements $\mathbf{D}_{t+1}^j = \mathbf{D}_t^j \cup (\mathbf{x}_t, y_t)$
    Update $\boldsymbol{\gamma}$ and $\mathbf{B}$ using EM algorithm in (15)
    if $t \,\%\, J = 0$ **then** $L = L/2$

Algorithm 3 LATSI

Assume: Sensing model (1); sparse signal β; J agents
Set: $\mathbf{D}_0^j = \varnothing$; $\boldsymbol{\tau}$: randomly initialized; α, η for tuning; p_0 : 1-sparse uniform distribution on $\boldsymbol{\beta}$
For $t = 1, ..., T$
  Wait for an agent to finish; **For** the free agent j:[2]
    Sample $\boldsymbol{\beta}^\star \sim p(\boldsymbol{\beta}|\mathbf{D}_t^j, \boldsymbol{\tau})$ from equation (10)
    Select $\mathbf{x}_t = \arg\max_{\mathbf{x}} \boldsymbol{R}^+(\boldsymbol{\beta}^\star, \mathbf{D}_t^j, \{\mathbf{x}, y\})$ from (33)
    Observe y_t given action $\mathbf{x}_t$
    Update & share measurements $\mathbf{D}_{t+1}^j = \mathbf{D}_t^j \cup (\mathbf{x}_t, y_t)$
    Update $\boldsymbol{\tau}$ using EM algorithm in (11)
    Update $p(\boldsymbol{\beta}|\mathbf{D}_t^j, \boldsymbol{\tau})$ using RSI-A [Ma et al., 2017]

expected regret $\mathbb{E}[Reg(T)] = \mathbb{E}\left[\sum_{t=1}^{T} \mathcal{R}(\mathbf{x}^\star, \boldsymbol{\beta}) - \mathcal{R}(\mathbf{x}_t, \boldsymbol{\beta})\right]$ *for 1-sparse and block-sparse algorithms are respectively upper-bounded by:*

1-sparse: $\mathbb{E}[Reg(T)] \leq$

$$\left(\log(|\mathcal{X}|) \sum_{t=1}^{\min\{T, n-1\}} \frac{\left(1 - \frac{t}{n}\right)\left(1 - \frac{1}{n-t+1}\right)}{\left(\frac{n-t-1}{n-t} \log\left(\frac{n-t}{n-t-1}\right) + \frac{1}{n-t} \log(n-t)\right)}\right)^{1/2} \quad (4)$$

Block-sparse: $\mathbb{E}[Reg(T)] \leq$

$$\left(\log(|\mathcal{X}|) \sum_{t=1}^{\min\{T, \log_2(n)\}} \left(1 - \frac{1}{n - \left(\sum_{t'=1}^{t-1} \frac{n}{2^{t'}}\right)}\right)^2 \Big/ \log(2)\right)^{1/2} \quad (5)$$

A simple comparison of (4) and (5) in Theorem 1 shows that using TS with a block sparse prior and varying block length significantly reduces the regret bounds comparing to TS that is using the true 1-sparse prior. Next, we will compute and compare an upper bound on the expected regret of a single-agent and an asynchronous multi-agent TS algorithm. To the best of our knowledge, only theoretical analysis for asynchronous parallel TS has been provided by Kandasamy et al. [2018] which is limited to Gaussian Processes. In the following theorem, we provide theoretical guarantees for an asynchronous multi-agent active search problem with a sparse model with proof in Appendix 5.

Theorem 2. *Consider the active search problem in Theorem 1. Let us propose two TS algorithms with a 1-sparse prior where one is single agent and another uses J agents in an asynchronous parallel setting. Then, the expected regret as defined in Theorem 1 for the single and multi-agent algorithms respectively are:*

$$\mathbb{E}[Reg(T)] = T_n - \frac{T_n(T_n+1)}{2n}, \quad T_n = \min\{T, n\} \quad (6)$$

$$\mathbb{E}[Reg(T)] \leq T_n - \frac{T_n(T_n+1)}{2n} + \frac{T_n(2J-1)}{n}, \; T_n = \min\{T, n+J\} \quad (7)$$

A simple analysis of (7) shows that for $J \ll n$ and $J \ll T$ (which is a reasonable assumption), the third term in the bound will be upper bounded by $2J+1$. As a result, the difference in expected regret between single agent and asynchronous multi-agent is negligible in terms of number of measurements T. Hence, we can conclude that by dividing the same number of measurements T between J agents, multi-agent algorithm achieves same regret J times faster than single agent setting.

Remark 2. *Theorem 2 shows that our asynchronous multi-agent algorithm performs on par with an optimal multi-agent system with a central planner. This result is a consequence of the central planner's regret being bounded by the single agent in terms of number of measurements T [Kandasamy et al., 2018].*

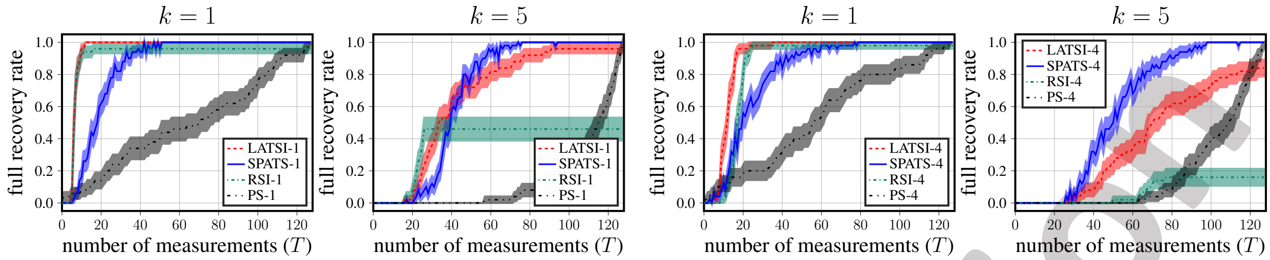
4.2 LATSI: LAPLACE TS WITH INFORMATION

In Section 3.2, we discussed how single agent Laplace-TS fails due to a careless assessment of the information gain. To combat this issue, Russo and Van Roy [2017] propose a new reward function which is a combination of expected regret and mutual information. They show that their algorithm called Information Directed Sampling (IDS) considerably improves the performance of single agent Laplace-TS. Unfortunately, computing the mutual information as introduced in IDS is not computationally feasible for our problem. Specifically, IDS proposes sampling to approximate the mutual information between the optimal action and the next observation. However, sampling the optimal action requires computing it for each sample which considering the region sensing assumption is quite expensive.

Another information-theoretic active search algorithm is Region Sensing Index(RSI) by Ma et al. [2017] that recognizes region sensing constraints. RSI searches for sparse signals by maximizing the mutual information between the

[1] The equations and derivations can be found in Appendix 3 in supplementary material

[2] The equations and derivations can be found in Appendix 1 and Appendix 6 in supplementary material



(a) single agent

(b) 4 agents

Figure 3: Full recovery rate of SPATS, LATSI, RSI and PS (exhaustive) for 1 and 4 agents for sparsity $k = 1, 5$

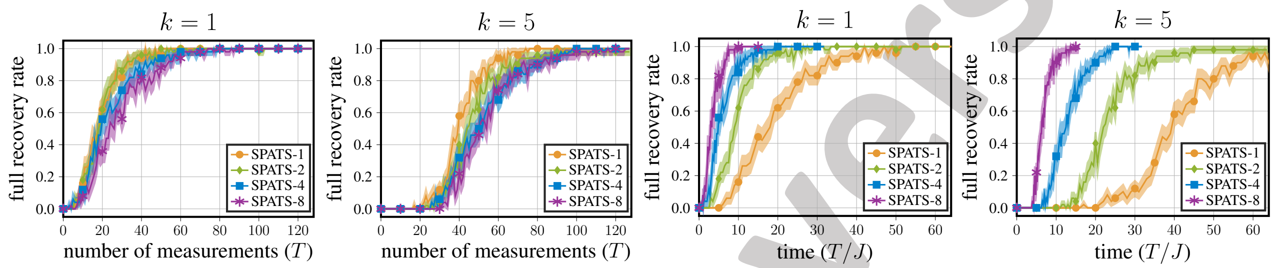


Figure 4: Full recovery rate of SPATS with 1, 2, 4 and 8 agents for sparsity rates $k = 1, 5$

next observation and the true parameter $\boldsymbol{\beta}$. RSI does not extend well to multi-agent settings as without randomness in its reward function, all agents will solve for the same sensing action. However, inspired by the IDS algorithm, we propose combining the reward of Laplace-TS with the mutual information computed in RSI. We call this algorithm LATSI summarized in Algorithm 3 with additional details in Appendix 6. We will provide simulation results in the next section, showing that while LATSI improves Laplace-TS, its performance is lower than SPATS for $k > 1$ due to RSI's poor approximation of mutual information there.

5 NUMERICAL RESULTS

We now compare the performance of our SPATS and LATSI against two methods 1) an information-theoretic approach called RSI proposed by Ma et al. [2017] 2) a point sensing (PS) approach that exhaustively searches the environment one location at a time. In this section, we focus on 2-dimensional search spaces ($d = 2$), where we estimate a k-sparse signal $\boldsymbol{\beta}$ with length $n = 8 \times 16$ and two sparsity rates of $k = 1, 5$. Here, $\boldsymbol{\beta}$ is generated with a randomly uniform sparse vector. We set the signal to noise variance to 16. For LATSI, we set the tuning parameters $\alpha, \eta = 1$. Note that neither SPATS nor LATSI are aware of the true uniform sparse prior or sparsity rate k. We then vary the number of measurements T and plot the mean and standard error of the full recovery rate over 50 random trials. The full recovery rate is defined as the rate at which an algorithm correctly recovers the entire vector $\boldsymbol{\beta}$ over random trials. To further demonstrate the efficacy of SPATS and LATSI, we provide additional experiments for $d = 1$, larger length n, $k = 10$, a sensitivity analysis for LATSI, and a robustness analysis to unreliability of communication for SPATS in Appendix 7.

Single-Agent

In a single agent setting, Figure 3a shows that for $k = 1$, RSI and LATSI outperform SPATS. The reason is that RSI has a very accurate approximation of mutual information for $k = 1$ and consequently it is difficult for our SPATS to win over the information-optimal algorithms of RSI and LATSI. All algorithms significantly outperform exhaustive search (PS). On the other hand, for higher sparsity rate of $k = 5$, SPATS outperforms RSI and LATSI. This is a result of poor approximation of mutual information for $k > 1$ by RSI. Specifically, for $k > 1$ RSI recovers the support of $\boldsymbol{\beta}$ by repeatedly applying RSI assuming $k = 1$. The authors use this strategy to avoid the large cost of computing mutual information for $k > 1$. This strategy even allows PS to catch up and outperform RSI. Finally, since our proposed LATSI is a combination of RSI and Laplace-TS, its performance is tied to that of both RSI and SPATS.

Multi-Agent

Figure 4, 5 and 6 show the performance of SPATS, RSI and LATSI in a multi-agent setting, respectively. Each figure consists of 4 sub-figures where the left two illustrate the full recovery rate for $k = 1$ and $k = 5$ as a function of number of measurements (T) taken by all the agents. To better demonstrate the multi-agent performance, on the right two subfigures we plot full recovery rate as a function of time which is computed by dividing the number of measurements

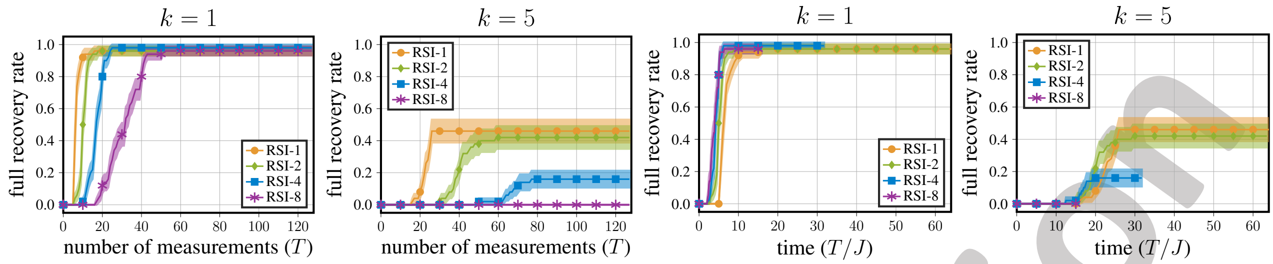


Figure 5: Full recovery rate of RSI with 1, 2, 4 and 8 agents for sparsity rates $k = 1, 5$

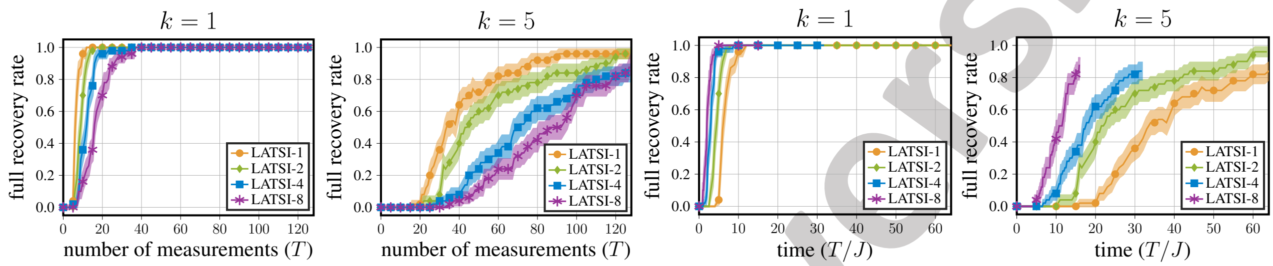


Figure 6: Full recovery rate of LATSI with 1, 2, 4 and 8 agents for sparsity rates $k = 1, 5$

T by the number of agents J. In each sub-figure we vary the number of available agents between $1, 2, 4$ and 8. Furthermore, in all subsequent plots, LATSI-J, RSI-J or SPATS-J indicate the corresponding algorithm with J agents.

SPATS: As evident in the right two sub-figures of Figure 4, SPATS becomes J times faster by using J number of agents. From the left two sub-figures, we can draw a similar conclusion. That is, increasing the number of agents from 1 to 2 to 4 and 8 hardly changes the total number of measurements required for a given recovery rate, i.e. the average number of sensing actions per agent is improved about J times. This result demonstrates that SPATS can efficiently perform active search in an asynchronous decentralized fashion.

RSI: We extend the RSI algorithm of Ma et al. [2017] to multi-agent setting by allowing each agent to independently choose its sensing action given RSI's acquisition function and utilizing the available measurements from other agents. Looking at Figure 5 for both $k = 1$ and $k = 5$, we see a significant deterioration in full recovery rate as a function of T as the number of agents increases. The reason is that without randomness in RSI's reward function, agents that are working at the same time are repeating the same sensing actions. For $k = 5$, this performance reduction is also obvious as a function of time. However, for $k = 1$ RSI performs slightly better in time by increasing agents. The reason for this contradicting behavior is that RSI's performance for $k = 1$ is so close to optimal (binary search) that it reaches recovery rate of 1 before the multi-agent system can negatively affect it.

LATSI: Looking at Figure 6, we see that similar to SPATS, LATSI's multi-agent performance improves in time by increasing the number of agents.

SPATS vs. LATSI vs. RSI: In Figure 3b, we plot all four algorithms against each other for 4 agents. Here, for $k = 1$, RSI and LATSI outperform SPATS due to their information-theoretic approach in computing the reward function. For $k = 5$, SPATS outperforms both RSI and LATSI. This is because SPATS is carefully designed to use randomness from TS in its reward function such that multiplying the number of agents would multiply its recovery rate. Furthermore, LATSI performs significantly better than RSI due to the probabilistic exploration aspect of TS in its reward function. All algorithms outperform PS except for RSI with $k > 5$ due to its poor information approximation. From our experiments, it is evident that LATSI is more suitable for scenarios where computing mutual information is cheap.

Lastly, the code for these experiments can be found at Ghods et al..

6 CONCLUSIONS

We have proposed two novel algorithms - SPATS and LATSI which are suitable for the recovery of sparse targets in a multi-agent (parallel) asynchronous active search problem with a region sensing constraint. We have discussed the role of sparsity in our design principle and also compared the limitations of a purely information theoretic approach in this setting. As part of a future work, we are currently working on considering traveling cost of targets by combining ideas from SPATS and LATSI with cost-lookahead strategies. We are also working on a collaboration applying our algorithms on real field robots. Another interesting direction for future work includes considering moving targets. One could also use continuous sensing along a trajectory as an objective rather than only sensing at stopping points.

References

Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. Online-to-confidence-set conversions and application to sparse stochastic bandits. In *AISTATS*, 2012.

Akash Arora, P Michael Furlong, Robert Fitch, Salah Sukkarieh, and Terrence Fong. Multi-modal active perception for information gathering in science missions. *Autonomous Robots*, 43(7):1827–1853, 2019.

Javad Azimi, Alan Fern, and Xiaoli Z Fern. Batch Bayesian optimization via simulation matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

Javad Azimi, Alan Fern, Xiaoli Zhang-Fern, Glencora Borradaile, and Brent Heeringa. Batch active learning via coordinated matching. *preprint arXiv:1206.6458*, 2012.

Kyounghwa Bae and Bani K Mallick. Gene selection using a two-level hierarchical Bayesian model. *Bioinformatics*, 20(18):3423–3430, 2004.

Graeme Best, Oliver M Cliff, Timothy Patten, Ramgopal R Mettu, and Robert Fitch. Dec-mcts: Decentralized planning for multi-robot active perception. *International Journal of Robotics Research (IJRR)*, 38(2-3), 2019.

Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

Gábor Braun, Sebastian Pokutta, and Yao Xie. Info-greedy sequential adaptive compressed sensing. *IEEE Journal of selected topics in signal processing*, 9(4):601–611, 2015.

Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2), 2006.

Alexandra Carpentier and Remi Munos. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. In *Artificial Intelligence and Statistics*, 2012.

Mattia Carpin, Stefano Rosati, Mohammad Emtiyaz Khan, and Bixio Rimoldi. UAVs using Bayesian optimization to locate wifi devices. *preprint arXiv:1510.03592*, 2015.

Benjamin Charrow, Nathan Michael, and Vijay Kumar. Cooperative multi-robot estimation and control for radio source localization. *IJRR*, 33(4), 2014.

J. Chen, T. Shu, T. Li, and C.W. de Silva. Deep reinforced learning tree for spatiotemporal monitoring with mobile robotic wireless sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, PP(99), 2019.

Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

Shih-Yi Chien, Huadong Wang, and Michael Lewis. Human vs. algorithmic path planning for search and rescue by robot teams. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 54(4), 2010.

Doo-Hyun Cho, Jung-Su Ha, Sujin Lee, Sunghyun Moon, and Han-Lim Choi. Informative path planning and mapping with multiple UAVs in wind fields. In *Distributed Autonomous Robotic Systems (DARS)*. Springer, 2018.

Oliver M Cliff, Robert Fitch, Salah Sukkarieh, Debra L Saunders, and Robert Heinsohn. Online localization of radio-tagged wildlife with an autonomous aerial robot system. In *Robotics: Science and Systems*, 2015.

Philip Dames, Pratap Tokekar, and Vijay Kumar. Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots. *IJRR*, 36(13), 2017.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11), 2004.

Mark A Davenport and Ery Arias-Castro. Compressive binary search. In *International Symposium on Information Theory Proceedings*, pages 1827–1831. IEEE, 2012.

David L Donoho. Compressed sensing. *IEEE Transactions on information theory (TIT)*, 52(4):1289–1306, 2006.

John T Feddema, Chris Lewis, and David Schoenwald. Decentralized control of cooperative robotic vehicles: theory and applications. IEEE, 2002.

Mário AT Figueiredo. Adaptive sparseness for supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 25(9):1150–1159, 2003.

Genevieve Flaspohler, Victoria Preston, Anna PM Michel, Yogesh Girdhar, and Nicholas Roy. Information-guided robotic maximum seek-and-sample in partially observable continuous environments. *IEEE Robotics and Automation Letters*, 4(4):3782–3789, 2019.

Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410), 1990.

Ramina Ghods, Arundhati Banerjee, and Jeff Schneider. Code for our SPATS and LATSI algorithms. *https://github.com/raminaghods/Multi-agent-active-search*.

Quanquan Gu, Tong Zhang, and Jiawei Han. Batch-mode active learning via error bound minimization. In *UAI*, pages 300–309, 2014.

Jarvis Haupt, Robert Nowak, and Rui Castro. Adaptive sensing for sparse signal recovery. In *13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, pages 702–707. IEEE, 2009a.

Jarvis D Haupt, Richard G Baraniuk, Rui M Castro, and Robert D Nowak. Compressive distilled sensing: Sparse recovery using adaptivity in compressive measurements. In *Asilomar Conference on Signals, Systems and Computers (Asilomar)*, pages 1551–1555. IEEE, 2009b.

Baichuan Huang, Jun Zhao, and Jingbin Liu. A survey of simultaneous localization and mapping. *arXiv preprint arXiv:1909.05214*, 2019.

Bruno Jedynak, Peter I Frazier, and Raphael Sznitman. Twenty questions with noise: Bayes optimal policies for entropy loss. *Journal of Applied Probability*, 49(1), 2012.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. Parallelised bayesian optimisation via thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, 2018.

Kirthevasan Kandasamy, Willie Neiswanger, Reed Zhang, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. Myopic posterior sampling for adaptive goal oriented design of experiments. In *ICML*, 2019.

Hung M La, Weihua Sheng, and Jiming Chen. Cooperative and active sensing in mobile sensor networks for scalar field mapping. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):1–12, 2014.

Mikko Lauri, Joni Pajarinen, and Jan Peters. Multi-agent active information gathering in discrete and continuous-state decentralized POMDPs by policy graph improvement. *Autonomous Agents & Multi-Agent Sys.*, 34, 2020.

John J Leonard and Hugh F Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *IROS*, volume 3, pages 1442–1447, 1991.

Pei Li and Haibin Duan. A potential game approach to multiple UAV cooperative search and surveillance. *Aerospace Science and Technology*, 68:403–415, 2017.

Zhan Wei Lim, David Hsu, and Wee Sun Lee. Adaptive informative path planning in metric spaces. *IJRR*, 2016.

Lanny Lin and Michael A Goodrich. Uav intelligent path planning for wilderness search and rescue. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, 2017.

Yifei Ma, Roman Garnett, and Jeff Schneider. Active search for sparse signals with region sensing. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Arian Maleki. Approximate message passing algorithms for compressed sensing. *a degree of Doctor of Philosophy, Stanford University*, 2011.

Matthew L Malloy and Robert D Nowak. Near-optimal adaptive compressed sensing. *IEEE TIT*, 60(7), 2014.

Roman Marchant and Fabio Ramos. Bayesian optimisation for intelligent environmental monitoring. In *IROS*, 2012.

Elaine Crespo Marques, Nilson Maciel, Lirida Naviner, Hao Cai, and Jun Yang. A review of sparse recovery algorithms. *IEEE Access*, 7:1300–1322, 2018.

Robin R Murphy. Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(2), 2004.

Deanna Needell and Joel A Tropp. CoSaMP: iterative signal recovery from incomplete and inaccurate samples. *Communications of the ACM*, 53(12):93–100, 2010.

Minh Tuan Nguyen. Distributed compressive and collaborative sensing data collection in mobile sensor networks. *Internet of Things*, page 100156, 2019.

Timothy Patten, Wolfram Martens, and Robert Fitch. Monte carlo planning for active object classification. *Autonomous Robots*, 42(2), 2018.

Jorge Peña Queralta, Jussi Taipalmaa, Bilge Can Pullinen, Victor Kathan Sarker, Tuan Nguyen Gia, Hannu Tenhunen, Moncef Gabbouj, Jenni Raitoharju, and Tomi Westerlund. Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *IEEE Access*, 8, 2020.

Purnima Rajan, Weidong Han, Raphael Sznitman, Peter Frazier, and Bruno Jedynak. Bayesian multiple target localization. *JMLR*, 37:1945–1953, 2015.

Cyril Robin and Simon Lacroix. Multi-robot target detection and tracking: taxonomy and survey. *Autonomous Robots*, 40(4):729–760, 2016.

Esther Rolf, David Fridovich-Keil, Max Simchowitz, Benjamin Recht, and Claire Tomlin. A successive-elimination approach to adaptive robotic sensing. *ArXiv*, 2018.

Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *Journal of Machine Learning Research (JMLR)*, 17(1), 2016.

Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. *Operations Research*, 66(1):230–252, 2017.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on Thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018.

Allison Ryan and J Karl Hedrick. A mode-switching path planner for uav-assisted search and rescue. In *IEEE Conference on Decision and Control*. IEEE, 2005.

Lorenzo Sabattini, Nikhil Chopra, and Cristian Secchi. Decentralized connectivity maintenance for cooperative control of mobile robotic systems. *IJRR*, 32(12), 2013.

Burr Settles. Active learning literature survey. Computer sciences technical report, University of Wisconsin–Madison, 2009.

Malcolm Strens. A Bayesian framework for reinforcement learning. In *ICML*, volume 2000, pages 943–950, 2000.

Hartmut Surmann, Rainer Worst, Tim Buschmann, Artur Leinweber, Alexander Schmitz, Gerhard Senkowski, and Niklas Goddemeier. Integration of UAVs in urban search and rescue missions. In *International Symposium on Safety, Security, and Rescue Robotics*. IEEE, 2019.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Michael E Tipping. Sparse Bayesian learning and the relevance vector machine. *JMLR*, 1(Jun), 2001.

Peter M Williams. Bayesian regularization and pruning using a Laplace prior. *Neural computation*, 7(1), 1995.

David P Wipf and Bhaskar D Rao. Sparse Bayesian learning for basis selection. *IEEE Transactions on Signal processing (TSP)*, 52(8):2153–2164, 2004.

Jeremy Wyatt. Exploration and inference in learning from reinforcement. 1998.

Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10(12):399, 2013.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *preprint arXiv:1911.10635*, 2019.

Zhilin Zhang and Bhaskar D Rao. Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):912–926, 2011.

Zhilin Zhang and Bhaskar D Rao. Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation. *IEEE TSP*, 61(8), 2013.