Randomized Sensing in Adversarial Environments

Andreas Krause
ETH Zürich and Caltech
Zürich, Switzerland
krausea@ethz.ch

Alex Roper
University of Michigan
Ann Arbor, MI, USA
aroper@umich.edu

Daniel Golovin
Caltech
Pasadena, CA, USA
dgolovin@caltech.edu

Abstract

How should we manage a sensor network to optimally guard security-critical infrastructure? How should we coordinate search and rescue helicopters to best locate survivors after a major disaster? In both applications, we would like to control sensing resources in uncertain, adversarial environments. In this paper, we introduce RSENSE, an efficient algorithm which guarantees near-optimal randomized sensing strategies whenever the detection performance satisfies submodularity, a natural diminishing returns property, for any fixed adversarial scenario. Our approach combines techniques from game theory with submodular optimization. The RSENSE algorithm applies to settings where the goal is to manage a deployed sensor network or to coordinate mobile sensing resources (such as unmanned aerial vehicles). We evaluate our algorithms on two real–world sensing problems.

1 Introduction

We consider the problem of optimizing sensing strategies in adversarial environments. Consider, for example, the problem of controlling Pan-Tilt-Zoom (PTZ) cameras in order to maximize the chance of detecting intrusions. Or consider the problem of planning paths for a team of search helicopters to locate survivors after a major disaster. In both problems, we have to choose among possible observations (where to point the cameras, where to send the helicopters), in order to maximize worst-case performance (protect against adversarial intrusions, or ensure uniformly high detection probability). These problems can be modeled as a game, where one player wishes to choose a set of observations, and the other player chooses a scenario (an intrusion or survivor location). Due to the combinatorial number of observation sets, however, the payoff matrix of this game is exponentially large. Fortunately, in many applications, for a fixed scenario the utility of the sensing player satisfies *submodularity* (*c.f.*, Fujishige, 2005), a natural diminishing returns property, which states that adding an observation helps more if we have made few observations, and less if we already have made many observations. If we have to commit to a fixed set of observations (e.g., deploy a set of static sensors), the goal then is to compute a (pure) minimax solution to this game. However, in many cases, we can avoid committing in advance and can play *randomized* strategies (*mixed* strategies in game–theory parlance), which are probability distributions over observation sets in this case. For example, we may wish to randomize the parameters of the PTZ cameras, or we may wish to randomize the surveillance paths taken by the mobile sensors. In such cases, the goal is to compute a minimax randomized strategy, which maximizes the expected performance against the worst-case scenario for it. In principle, such randomized strategies can provide arbitrarily better expected performance than deterministic strategies.

In this paper, we develop an efficient algorithm, RSENSE, for computing such randomized strategies in matrix games, whenever the performance measure of the sensing player is submodular. More generally, we show that whenever one of the players uses an approximation algorithm (such as an existing algorithm for constrained submodular maximization) to compute best responses in the game, this algorithm can be used to efficiently compute randomized strategies while preserving the approximation guarantee of the player's algorithm. The algorithm applies under a variety of natural objective functions (such as variance reduction, or probability of detection), and constraints (e.g., selecting the best k sensor locations; controlling PTZ cameras; or planning paths for mobile sensors with constraints on the path lengths). In many cases, optimizing for the worst-case may be too pessimistic. We thus also consider the problem of trading off worst-case and average-case (with respect to some known prior distribution) performance, and develop an efficient algorithm, TRSENSE, for optimizing this tradeoff. We empirically evaluate our algorithms on two real-world sensing tasks: The problem of choosing sensing locations to minimize worst-case prediction error in environmental monitoring, and the problem of coordinating search and rescue helicopters to maximize the worst-case probability of detecting survivors after a major disaster. Our results indicate, among other things, that randomized sensing strategies can dramatically outperform deterministic strategies obtained using existing techniques, and that effective tradeoffs between worst-case and average-case performance can be achieved.

2 The Randomized Sensing Problem

Suppose we would like to monitor an environment, discretized into a finite set $\mathcal{V}$ of locations. We can obtain observations from a subset $\mathcal{A} \subseteq \mathcal{V}$ of those locations in order to detect

adversarial incursions. Suppose the adversary has a finite set $\mathcal{I}$ of strategies for possible incursions. With any given strategy $i \in \mathcal{I}$, we associate a sensing quality objective function $F_i : 2^{\mathcal{V}} \to [0, 1]$ that models whether, and how well, observations at locations $\mathcal{A}$ would allow us to detect incursion i. For example, an incursion i could be associated with a set of locations $\mathcal{B}_i$ that the adversary must traverse, and the sensing quality could be one if the adversary traverses a location with a sensor and zero otherwise, i.e., $F_i(\mathcal{A}) = \min\{1, |\mathcal{B}_i \cap \mathcal{A}|\}$. We give other concrete examples of sensing quality functions in Section 4.1. However, note that F_i encodes everything we need to know about i, and allows us to abstract away the application–specific details of what constitutes an incursion under certain assumptions. Specifically, we assume, w.l.o.g., that $F_i(\emptyset) = 0$ for all i, i.e., zero sensors provide no utility, and also that F_i is scaled so that $F_i(\mathcal{A}) \leq 1$ for all $\mathcal{A}$. Furthermore, we assume that each F_i is *monotonic*, i.e., $F_i(\mathcal{A}) \leq F_i(\mathcal{B})$ whenever $\mathcal{A} \subseteq \mathcal{B}$. Thus, adding sensors can only help. Many natural sensing quality functions, such as those considered in Section 4.1, satisfy an additional, natural property called *submodularity*: For any $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and $s \in \mathcal{V} \setminus \mathcal{B}$ it holds that $F_i(\mathcal{A} \cup \{s\}) - F_i(\mathcal{A}) \geq F_i(\mathcal{B} \cup \{s\}) - F_i(\mathcal{B})$. Thus adding a new observation helps more if we have made few observations so far, and less if we have already made many observations.

First consider the setting where there is only one possible incursion strategy $i \in \mathcal{I}$. In this case, we wish to find, for example, the set $\mathcal{A}^*$ of k locations to observe that maximizes the sensing quality $F_i(\mathcal{A})$, i.e., $\mathcal{A}^* = \arg\max_{|\mathcal{A}| \leq k} F_i(\mathcal{A})$. This optimization problem is NP-hard for most interesting classes of objective functions. However, a seminal result of Nemhauser *et al.* [1978] proves that a simple greedy algorithm, that starts with the empty set $\mathcal{A}_0 = \emptyset$ and iteratively adds the element maximizing the improvement in value,

$$\mathcal{A}_{\ell+1} = \mathcal{A}_\ell \cup \{\arg\max_{s \in \mathcal{V} \setminus \mathcal{A}_\ell} F_i(\mathcal{A}_\ell \cup \{s\})\},$$

achieves a near-optimal solution: It holds that $F_i(\mathcal{A}_k) \geq (1 - 1/e) \max_{|\mathcal{A}| \leq k} F_i(\mathcal{A})$. Moreover, under reasonable complexity theoretic assumptions, no polynomial time algorithm can provide a $(1 - 1/e + \varepsilon)$ guarantee for any $\varepsilon > 0$ [Feige, 1998]. This insight has been exploited in sensor placement and information gathering [Krause and Guestrin, 2007].

Now suppose there is more than one possible incursion strategy ($|\mathcal{I}| > 1$). If we have to commit to a fixed set of locations (e.g., by deploying security cameras), then the adversary gets to see these locations and may pick the worst possible intrusion for them. Thus, our goal would be to pick the set

$$\mathcal{A}^* = \arg\max_{|\mathcal{A}| \leq k} \min_{i \in \mathcal{I}} F_i(\mathcal{A}). \tag{1}$$

Krause *et al.* [2008] show that this problem is extremely intractable: Under reasonable complexity assumptions, it is not possible to achieve *any* approximation to this problem, i.e., for any function $g(n)$ that can depend on the size n of the problem instance, it is not possible to efficiently obtain a solution $\mathcal{A}'$ such that $\min_i F_i(\mathcal{A}') \geq g(n) \max_{|\mathcal{A}| \leq k} \min_i F_i(\mathcal{A})$.

Now instead of deploying a fixed set of sensors in advance, suppose that we can *randomize*: For example, we have deployed a network of PTZ cameras, and at every time-step we wish to randomly point each camera in a particular direction. Or we have mobile sensors (e.g., search helicopters), and assign random locations / routes to these sensors. We presume the adversary will eventually learn this distribution (e.g., by watching how our sensing assets are used over time). In this case, the goal becomes to obtain a *distribution* P over sets such that even an intrusion optimized against this distribution does as little damage as possible, i.e., we are interested in

$$P^* = \arg\max_{P \in \mathcal{P}} \min_{i \in \mathcal{I}} \sum_{\mathcal{A}} P(\mathcal{A}) F_i(\mathcal{A}) \tag{2}$$

where $\mathcal{P}$ is the set of all distributions whose support contains only feasible observation sets. Note we could allow the adversary to randomize over incursions, however because the adversary gets to select i after we select P, the adversary derives no benefit from randomization. In contrast, with randomization we can do arbitrarily better than deploying a fixed set of sensors: Suppose there are two incursions $\mathcal{I} = \{1, 2\}$, and two locations, $\mathcal{V} = \{1, 2\}$. The sensing quality functions are such that $F_i(\mathcal{A}) = 1$ iff $i \in \mathcal{A}$ and 0 otherwise. Suppose we are allowed to pick one observation. In this case, it can be seen that $\max_{|\mathcal{A}| \leq 1} \min_i F_i(\mathcal{A}) = 0$, whereas $\max_{P \in \mathcal{P}} \min_{i \in \mathcal{I}} \sum_{\mathcal{A}} P(\mathcal{A}) F_i(\mathcal{A}) = \frac{1}{2}$.

However, solving Problem (2) is a formidable task: The optimal distribution might have exponentially large support. Perhaps surprisingly, in this paper we show how we can, for any $\varepsilon > 0$, efficiently find a distribution P' such that

$$\mathbf{V}(P') \geq (1 - 1/e) \max_{P \in \mathcal{P}} \mathbf{V}(P) - \varepsilon,$$

where $\mathbf{V}(P) = \min_i \sum_{\mathcal{A}} P(\mathcal{A}) F_i(\mathcal{A})$ is the sensing quality of distribution P in the worst case. Note that, in contrast to Problem (1) which is extremely inapproximable, for Problem (2) we can give essentially the same approximation guarantees as for the classical, non-adversarial case, up to some absolute error ε that can be made arbitrarily small.

In fact, this result can be generalized. In many applications, we have more complex constraints than simply choosing the best k sensor locations. For example, we may have deployed a network of PTZ cameras, and must choose one pan, tilt and zoom setting for each camera. Or we may have mobile sensors (e.g., security personnel) and must choose patrols (i.e., paths to move along while observing) of bounded length for each one. More generally, we assume that we would like to pick a distribution P over sets $\mathcal{A}$ such that each set $\mathcal{A}$ satisfies some constraints, i.e., $\mathcal{A} \in \mathcal{C}$ for some constraint set $\mathcal{C} \subseteq 2^{\mathcal{V}}$. We call the following problem the *randomized submodular sensing problem*:

$$P^* = \arg\max_{P : P(\mathcal{A}) > 0 \Rightarrow \mathcal{A} \in \mathcal{C}} \mathbf{V}(P). \tag{3}$$

In the following, we will show that we can (approximately) solve Problem (3) whenever we can (approximately) solve the problem $\max_{\mathcal{A} \in \mathcal{C}} F(\mathcal{A})$ for any monotonic submodular function F. For both applications mentioned above (control of PTZ cameras and planning paths for mobile sensors), approximation algorithms for the problem $\max_{\mathcal{A} \in \mathcal{C}} F_i(\mathcal{A})$ are known [Vondrák, 08; Singh *et al.*, 2009]. We provide detailed examples of randomized sensing problems in Section 4.1.

3 The RSENSE Algorithm

We now describe our approach towards Problem (3). The key idea is to consider P as a mixed strategy (i.e., a distribution over pure strategies) in an (exponentially-large) zero-sum matrix game, where the rows enumerate all feasible sensor placements $\mathcal{A}$, the columns enumerate all intrusion strategies i, and the matrix entry $\mathbf{M}(\mathcal{A}, i)$ has value $F_i(\mathcal{A}) \in [0, 1]$. For distributions P over placements and Q over incursions, we write $\mathbf{M}(P, Q) = \sum_{\mathcal{A}} \sum_i P(\mathcal{A})Q(i)F_i(\mathcal{A})$, and set $\mathbf{V}(P) = \min_Q \mathbf{M}(P, Q)$ as the performance of P against an adversarially chosen randomized[1] incursion.

In a seminal result, Freund and Schapire [1999] show that a simple multiplicative update algorithm can be used to approximate optimal mixed strategies in an (arbitrary) matrix game, as long as 1) one of the players has a small (polynomial size) number of choices, and 2) the other player can compute *best responses*, i.e., can compute $\max_P \mathbf{M}(P, Q)$ for any distribution Q. In the following, we review this algorithm, adapted to the context (and using the notation) of our application.

This iterative algorithm generates a sequence of distributions Q_t and best responses P_t. Here, Q_1 is the uniform distribution. In each iteration t, P_t is the best response to Q_t,

$$P_t = \arg\max_P \mathbf{M}(P, Q_t),$$

and given P_t, Q_{t+1} is computed by

$$Q_{t+1}(i) = Q_t(i)\frac{\beta^{\mathbf{M}(P_t,i)}}{Z_t},$$

where $Z_t = \sum_i Q_t(i)\beta^{\mathbf{M}(P_t,i)}$ is a normalization factor, $\beta = 1/(1+\sqrt{2\ln|\mathcal{I}|/T})$, and T is a specified bound on the number of iterations. Freund and Schapire [1999] show[2] that the *average* best response, $\bar{P} = \sum_{t=1}^T P_t$, satisfies

$$\mathbf{V}(\bar{P}) \geq \max_P \mathbf{V}(P) - \Delta_{T,\mathcal{I}},$$

where $\Delta_{T,\mathcal{I}} = \sqrt{\frac{2\ln|\mathcal{I}|}{T}} + \frac{\ln|\mathcal{I}|}{T} = \mathcal{O}\left(\sqrt{\frac{\ln|\mathcal{I}|}{T}}\right)$. Thus, $T = \Theta\left(\frac{\log|\mathcal{I}|}{\varepsilon^2}\right)$ iterations suffice to ensure $\Delta_{T,\mathcal{I}} = \mathcal{O}(\varepsilon)$.

It takes $\mathcal{O}(|\mathcal{I}|)$ evaluations of $\mathbf{M}$ to obtain Q_{t+1} from Q_t and P_t. Let us assume that the number of intrusion scenarios $|\mathcal{I}|$ is polynomially bounded and $\mathbf{M}$ can be efficiently evaluated. Then in order to apply the above algorithm to our setting, the key problem is computing the best response P_t. It can be seen that we can restrict ourselves to considering only deterministic strategies, i.e., those that put all probability mass on a single set of sensor locations $\mathcal{A}_t$ (formally $P_t = \delta_{\mathcal{A}_t}$, where $\delta_{\mathcal{A}}$ is the Dirac delta function which equals one if its argument is $\mathcal{A}$ and zero otherwise). Thus, to compute the best response, we need to solve the problem

$$\mathcal{A}_t = \arg\max_{\mathcal{A}\in\mathcal{C}} \sum_i Q_t(i)F_i(\mathcal{A}). \quad (4)$$

Since non-negative linear combinations of monotonic submodular functions remain monotonic submodular, computing the best response (4) requires solving a constrained submodular maximization problem. As discussed in Section 2, for most interesting objective functions this is an NP-hard problem, but fortunately there exist approximation algorithms for a variety of constraints. Now the key question is: Suppose we use an α-approximation algorithm for solving Problem (4), and thus each P_t is an *α-best response* to distribution Q_t. What does this imply for the randomized submodular sensing problem (3)? It may well be possible that simply using α-best responses can lead to arbitrarily poor performance. In fact, exactly this is known to be the case when naively using an α-approximate algorithm in online optimization [Kalai and Vempala, 2005]. Fortunately, and perhaps surprisingly given the similarity of online optimization and computing equilibria in matrix games, this is not the case. We call the resulting algorithm RSENSE, and show:

Theorem 1. *Suppose we have an algorithm that, given monotone submodular function $F : 2^{\mathcal{V}} \to [0, 1]$ and constraint system $\mathcal{C}$, will find $\mathcal{A}' \in \mathcal{C}$ such that $F(\mathcal{A}') \geq \alpha \max_{\mathcal{A}\in\mathcal{C}} F(\mathcal{A})$ for some $\alpha > 0$. Fix $\varepsilon > 0$ and let $T = 4\lceil\frac{\ln|\mathcal{I}|}{\varepsilon^2}\rceil$. Then* RSENSE*, run for T iterations with this subroutine, produces a sequence $\mathcal{A}_1, \ldots, \mathcal{A}_T$ such that the average distribution $\bar{P} = \frac{1}{T}\sum_{t=1}^T \delta_{\mathcal{A}_t}$ satisfies*

$$\mathbf{V}(\bar{P}) \geq \alpha \max_{P:P(\mathcal{A})>0\Rightarrow\mathcal{A}\in\mathcal{C}} \mathbf{V}(P) - \varepsilon.$$

The proof of this theorem will follow from a more general result proved in the next section. In the case of both RSENSE and the TRSENSE algorithm of §3.1, the running time is dominated by the time to make $\mathcal{O}(\frac{\ln|\mathcal{I}|}{\varepsilon^2})$ calls to the α-best response subroutine, plus the time to evaluate the payoff matrix $\mathbf{M}$ a total of $\mathcal{O}(|\mathcal{I}|\ln|\mathcal{I}|/\varepsilon^2)$ times.

3.1 Trading Off Worst-Case and Average-Case Performance

In many cases, assuming adversarial incursions may be overly pessimistic. For example, when monitoring a public property with security cameras, we may simultaneously want to protect the property against incursions, while also ensuring that we achieve good coverage "on average". In such settings, we may both care about the sensing quality of distribution P in the worst case (i.e., $\mathbf{V}(P)$), and about $\mathbf{M}(P, R)$, for some specified distribution R, e.g., the uniform distribution, or a distribution that puts more weight on certain incursions. Thus, a natural problem is to trade off these two quantities, which we can do using the scalarized objective

$$\mathbf{V}_\lambda(P) := \lambda\mathbf{V}(P) + (1-\lambda)\mathbf{M}(P, R), \quad (5)$$

using tradeoff parameter $\lambda \in [0, 1]$. Setting $\lambda = 1$ means we only care about worst-case and $\lambda = 0$ means we only care about average-case performance. Note that since the function $F(\mathcal{A}) = \mathbf{M}(\mathcal{A}, R)$ is submodular, the setting $\lambda = 0$ is just the classical problem of constrained submodular maximization.

A natural approach towards extending our RSENSE algorithm for this tradeoff is to optimize the best response not with respect to the distribution Q_t, but with respect to $\lambda Q_t + (1-\lambda)R$, i.e., setting

$$P_t = \arg\max_P \mathbf{M}(P, \lambda Q_t + (1-\lambda)R).$$

[1] Note this minimum can be achieved by some pure strategy i, so that $\mathbf{V}(P) = \min_{i\in\mathcal{I}} \mathbf{M}(P, i)$.

[2] See their proof of von Neumann's minimax theorem.

We call this generalization the TRSENSE algorithm, and prove the following result:

Theorem 2. *Suppose we are given an arbitrary distribution R, $\lambda \in [0,1]$, and an algorithm meeting the conditions of Theorem 1 (i.e., one which finds α-best responses). Fix $\varepsilon > 0$ and let $T = 4\lceil \frac{\ln |\mathcal{I}|}{\varepsilon^2} \rceil$. Then* TRSENSE*, run for T iterations with this subroutine, produces a sequence $\mathcal{A}_1, \ldots, \mathcal{A}_T$ such that the average distribution $\bar{P} = \frac{1}{T}\sum_{t=1}^{T} \delta_{\mathcal{A}_t}$ satisfies*

$$\mathbf{V}_\lambda(\bar{P}) \geq \alpha \max_{P: P(\mathcal{A})>0 \Rightarrow \mathcal{A} \in \mathcal{C}} \mathbf{V}_\lambda(P) - \varepsilon.$$

Proof. Let $v = \max_P \min_Q \mathbf{M}(P, \lambda Q + (1-\lambda)R)$. Then

$$\begin{aligned}
v &\leq \frac{1}{T}\sum_t \max_P \mathbf{M}(P, \lambda Q_t + (1-\lambda)R) && (6)\\
&\leq \frac{1}{T}\sum_t \frac{1}{\alpha}\mathbf{M}(P_t, \lambda Q_t + (1-\lambda)R) && (7)\\
&\leq \left(\frac{1}{\alpha T}\min_Q \sum_t \mathbf{M}(P_t, \lambda Q + (1-\lambda)R)\right) + \frac{1}{\alpha}\Delta_{T,\mathcal{I}} && (8)\\
&= \frac{1}{\alpha}\min_Q \mathbf{M}(\bar{P}, \lambda Q + (1-\lambda)R) + \frac{1}{\alpha}\Delta_{T,\mathcal{I}} && (9)\\
&= \frac{1}{\alpha}\mathbf{V}_\lambda(\bar{P}) + \frac{1}{\alpha}\Delta_{T,\mathcal{I}}. && (10)
\end{aligned}$$

Hereby, (6) holds since Q_t is a feasible (but not necessarily optimal) distribution; (7) holds since we use an α-approximation algorithm to compute P_t; (8) holds since it is (up to scaling by $1/\alpha$) a restatement of the fact that the multiplicative weight algorithm (as played by our simulation of the adversary) has regret bound $\Delta_{T,\mathcal{I}}$ [Freund and Schapire, 1999]; (9) holds due to the definition of $\bar{P}$ and $\mathbf{M}$; and (10) holds by the definition of $\mathbf{V}_\lambda$. □

4 Experimental Results

We perform experiments on two case studies. The questions we wish to answer are:

1. What is the potential benefit of selecting a distribution over observation sets rather than a fixed set?
2. What kind of distributions $P(\mathcal{A})$ does RSENSE choose?
3. How efficient is the RSENSE algorithm?
4. How does RSENSE compare to prior work?
5. How do average and worst-case performance trade-off?

4.1 Data Sets and Experimental Setup

Environmental Monitoring. We conduct out first set of experiments on an environmental monitoring task. The goal is to use robotic sensors to monitor spatial phenomena. In particular, we consider the problem of monitoring acidity of a lake near the University of California, Merced, using the NIMS-RD robot developed by Harmon *et al.* [2006]. This robot is suspended and can move along a horizontal transect of the lake, as well as lower and raise its sensing unit to obtain pH measurements at particular locations and depths. We discretize a horizontal transect of the lake into a finite set of 86 locations $\mathcal{V}$. With each location s, we associate a random variable $\mathcal{X}_s$, and model the joint probability distribution over the random vector $\mathcal{X}_\mathcal{V}$ using a nonstationary Gaussian process (GP) model of the phenomenon, estimated from data provided by Krause *et al.* [2008]. Given measurements $\mathcal{X}_\mathcal{A} = \mathbf{x}_\mathcal{A}$ at a subset $\mathcal{A} \subseteq \mathcal{V}$ of locations, this model allows us to quantify the predictive variance $\mathrm{Var}(\mathcal{X}_i \mid \mathcal{X}_\mathcal{A} = \mathbf{x}_\mathcal{A})$ for the pH value $\mathcal{X}_i$ at any given location i. Our goal is to choose a set $\mathcal{A}^*$ of k measurement locations in order to minimize the worst-case prediction error $\max_i \{\mathrm{Var}(\mathcal{X}_i \mid \mathcal{X}_\mathcal{A})\}$, where $\mathrm{Var}(\mathcal{X}_i \mid \mathcal{X}_\mathcal{A}) = \mathbb{E}_{\mathbf{x}_\mathcal{A}}[\mathrm{Var}(\mathcal{X}_i \mid \mathcal{X}_\mathcal{A} = \mathbf{x}_\mathcal{A})]$ is the expected squared prediction error. In our model with constant prior variance $\mathrm{Var}(\mathcal{X}_i) = const$, this problem is equivalent to maximizing the worst-case *variance reduction* $\min_i F_i(\mathcal{A})$, where

$$F_i(\mathcal{A}) = \mathrm{Var}(\mathcal{X}_i) - \mathrm{Var}(\mathcal{X}_i \mid \mathcal{X}_\mathcal{A}).$$

F_i is monotonic, and $F_i(\emptyset) = 0$. Furthermore, as shown by Das and Kempe [2008], under some conditions on the GP distribution, each function F_i is submodular.

Urban Search and Rescue. Our second set of experiments is on a problem of managing static and mobile sensors for the purposes of urban search and rescue (S&R). We take a map of the city of Southampton, as provided as part of the Robocup Rescue Challenge[3], and illustrated in Figure 3(a). We discretize the environment into 500 possible measurement locations $\mathcal{V}$, and 6500 points $\mathcal{I}$ that we would like to monitor. For each possible measurement location $s \in \mathcal{V}$, we use a line-of-sight algorithm to determine the locations R_s that can be seen from s. Suppose we choose a set $\mathcal{A}$ of observations. We assume that sensors are noisy, thus for a sensor s and location $i \in R_s$, with probability p, s fails to observe i. We choose $p = 0.2$. Suppose each sensor failure occurs independently. Then the chance of detecting a survivor at location i is

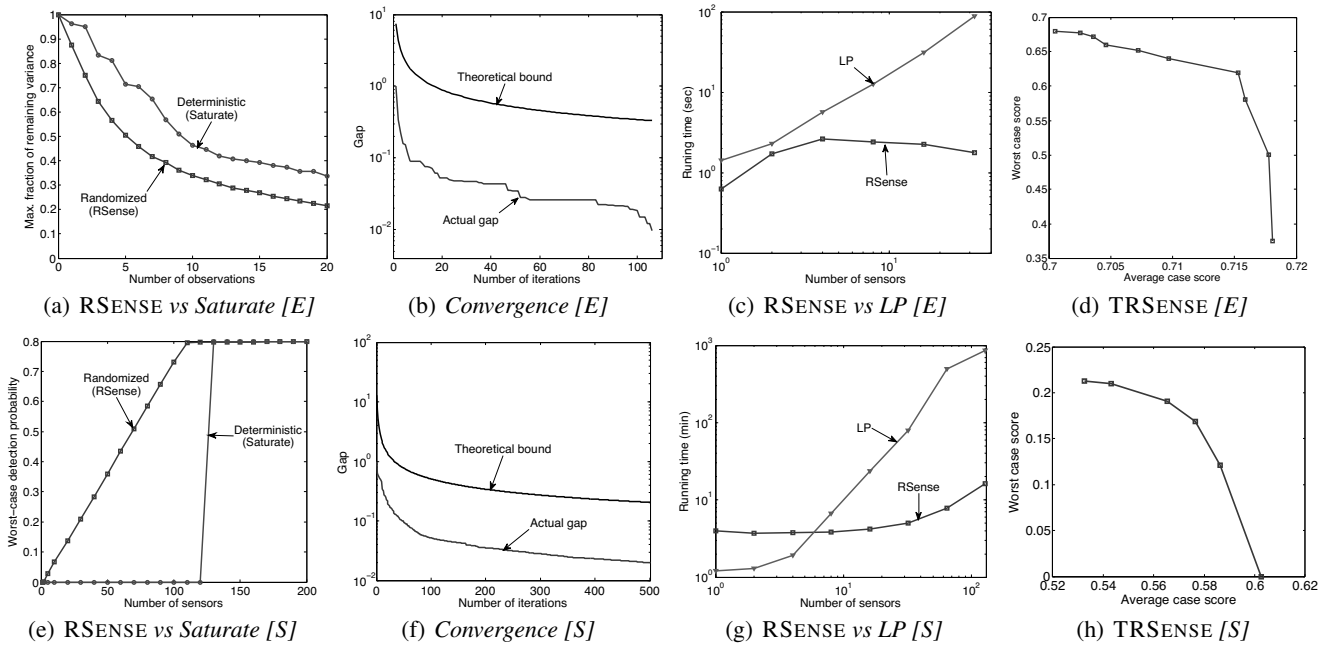
$$F_i(\mathcal{A}) = 1 - p^{|\{s \in \mathcal{A}: i \in R_s\}|}.$$

It can be shown that each $F_i(\mathcal{A})$ is monotonic and submodular. More complex models can be handled as well. E.g., the failure probability may depend on the distance of i and s, and the sensor failures may be correlated in a complex manner. While these extensions can be handled by our framework, we do not discuss them further due to space limitations.

4.2 Results

Random vs. Deterministic Strategies. We first wish to understand, how much better, in expectation, we can perform if we are allowed to choose a distribution $P(\mathcal{A})$ over measurement sets $\mathcal{A}$ instead of committing to a fixed set $\mathcal{A}$. The problem of choosing a fixed set $\mathcal{A}$ was studied by Krause *et al.* [2008], who developed the SATURATE algorithm, a bicriterion algorithm (with optimal approximation guarantees under reasonable complexity assumptions) for Problem (1). In this paper, we compare our RSENSE algorithm (which is allowed to pick distributions $P(\mathcal{A})$) with this existing approach. Figure 1(a) compares the (expected) worst-case remaining variance achieved by both algorithms on the environmental monitoring problem. It can be seen that RSENSE can exploit the power of randomization to achieve less predictive variance than SATURATE, which produces a fixed set (i.e., a

[3] http://www.robocuprescue.org/



(a) RSENSE *vs Saturate [E]* (b) *Convergence [E]* (c) RSENSE *vs LP [E]* (d) TRSENSE *[E]*

(e) RSENSE *vs Saturate [S]* (f) *Convergence [S]* (g) RSENSE *vs LP [S]* (h) TRSENSE *[S]*

Figure 1: Results on environmental monitoring [E] and search and rescue [S]. (a,e) Randomized strategies outperform deterministic strategies. (b,f) Observed convergence matches theory. (c,g) RSENSE scales better with larger numbers of sensors. (d,h) Tradeoff curves exhibit a strong "knee".

deterministic strategy). For example, with 10 observations, RSENSE achieves approximately the same variance reduction as SATURATE with 20 observations.

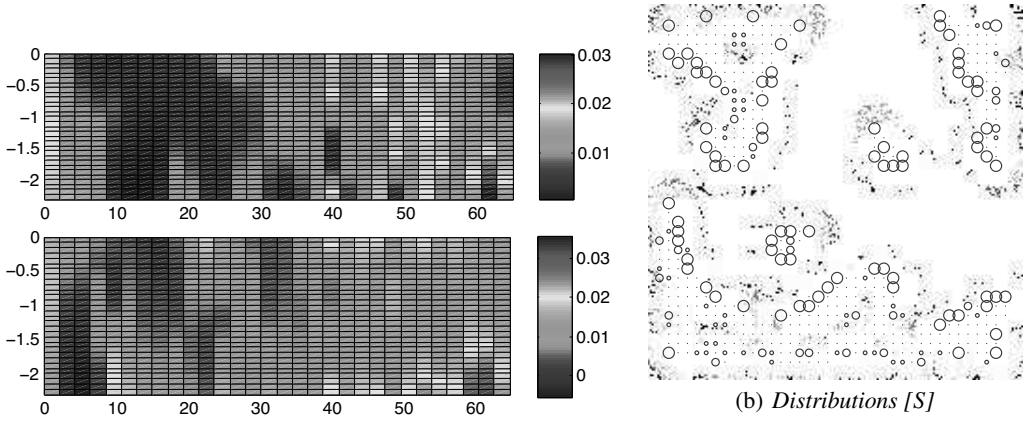
Figure 1(e) presents the worst-case detection probability for the search and rescue task. Interestingly, unless more than 120 sensors are deployed, the worst-case score of the deterministic solution stays zero, as there remain some locations that fewer sensors must leave unobserved. On the other hand, the performance of the RSENSE solution increases approximately linearly, until a near-maximal performance is obtained already using only 100 measurements.

Illustration of Distributions. Figure 2(a) visualizes the worst-case distribution Q over functions F_i (top), and the final distribution over sensing locations obtained by RSENSE (bottom), for the environmental monitoring problem. For the sensing plot, we visualize the marginal probability of whether an observation is made at each location and depth. Notice the distributions are non-trivial due to the non-stationarity of the GP distribution, i.e., the fact that the decay in spatial correlation is not translation invariant: The monitored phenomenon is "rougher" (less spatially correlated) for larger coordinates. Thus, more observations are needed to achieve low predictive variance.

Figure 2(b) presents the distributions for the S&R task. Notice that the worst-case distribution Q over objectives F_i (black dots) puts most probability mass close to the boundaries of buildings, which are difficult to observe due to the line-of-sight constraints. Little probability mass (lighter gray) is put in the open spaces that are easy to observe. Notice that the sensor distribution (blue circles) puts more probability mass (larger radius) away from boundaries, preferring instead locations where each measurement can observe multiple boundary points. Also note that both distributions are rather sparse.

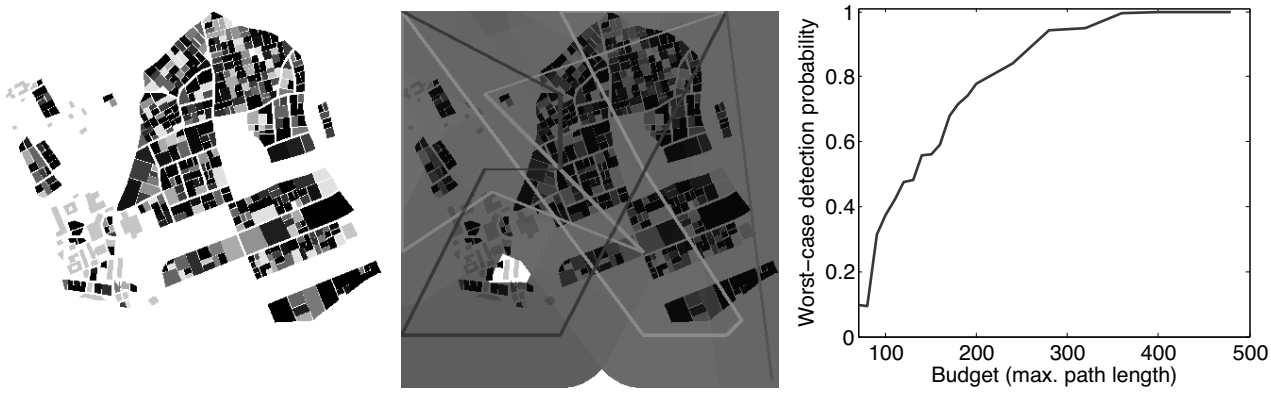
Convergence. Figure 1(b) illustrates the convergence properties of RSENSE for environmental monitoring. For a fixed number of 10 sensors, we plot the (empirical) gap between an (approximate due to NP-hardness) upper bound and a lower bound on the optimal sensing quality. Specifically, the gap is computed as $\mathbf{M}(P', \bar{Q}) - \mathbf{V}(\bar{P})$ where P' is an α-best response to the "average incursion distribution" $\bar{Q} = \frac{1}{T}\sum_t Q_t$, and $\bar{P} = \frac{1}{T}\sum_t P_t$. We also plot the theoretical bound $\varepsilon(T) = 2\sqrt{\ln|\mathcal{I}|/T}$ from Theorem 1. Notice that the empirical gap qualitatively behaves as predicted by the theory, even though the bound is loose by some constant factor. Figure 1(f) shows the same experiment for the S&R problem, which exhibits qualitatively similar behavior.

Running Time Comparison with Prior Art. Next, we compare RSENSE with a linear programming (LP) based double-oracle algorithm that has been proposed by Halvorson *et al.* [2009] for a similar problem (as reviewed in more detail in Section 5). Their algorithm also relies on a best-response oracle, and we use the same as for RSENSE. For a fixed error tolerance of at most 1% of variance, Figure 1(c) compares how the running time of both algorithms depends on the number of observations for environmental monitoring. Notice that RSENSE has a very weak dependence on this number, in contrast to the LP-based algorithm, which has orders of magnitude larger running time for larger numbers of sensors (and could require exponentially many iterations in the worst case). Figure 1(g) shows the same experiment on the S&R



(a) *Distributions [E]* (b) *Distributions [S]*

Figure 2: (a) Sensor (bottom) and adversarial (top) distributions in the environmental monitoring task. The x-axis corresponds to the location along the transect, y-axis corresponds to depth. The distributions are non-uniform, reflecting the nonstationarity of the phenomenon. Locations with larger x-coordinates have stronger decay of the spatial correlation, thus are harder to predict, and more measurements are obtained there. (b) Sensor (blue circles) and adversarial (black dots) distributions in the S&R task. Note the sparsity of the distributions. Most probability mass is concentrated around boundaries of buildings which are harder to observe than open spaces.



(a) *S&R height map [S]* (b) *Paths [S]* (c) *Path value [S]*

Figure 3: (a) Southampton height map used in the S&R task (darker is higher). (b) Three examples of paths sampled from the distribution $P(\mathcal{A})$. (c) Worst-case detection probability depending on path length.

problem. Note that while for small numbers of sensors the LP solution is faster, for larger numbers of sensors, the LP running time is orders of magnitudes larger than that of RSENSE. For example, when selecting 64 observations, RSENSE takes approximately 8 minutes, compared to approximately 8 hours for the LP solution.

Trading Off Average-Case and Worst-Case Performance. We also use our TRSENSE algorithm to optimize the tradeoff between average and worst case. Hereby, we use the uniform distribution over the open space to evaluate the average case performance. We vary the tradeoff parameter λ between 0 and 1 and plot the worst-case and average-case scores obtained for all the solutions. Figure 1(d) presents the results for environmental monitoring (using 10 observations), Figure 1(h) presents the results for S&R. Both tradeoff curves exhibit a strong knee; there are solutions which attain near-optimal scores *simultaneously* with respect to the average-case and worst-case scores.

Informative Path Planning. For the S&R problem, we also consider the problem of planning paths for mobile sensors, rather than choosing fixed sets of sensor locations. In this case, the feasible sets $\mathcal{C}$ are all paths of a given maximum length. Maximizing submodular functions subject to such constraints is a much more difficult optimization problem than choosing k arbitrary locations, and the greedy algorithm can potentially perform arbitrarily badly [Singh *et al.*, 2009]. Instead, we use an approximation algorithm developed by Singh *et al.* [2009] to compute best responses. Figure 3(b) illustrates samples from the distribution over paths chosen by RSENSE. Note how the paths route around the high ground to maximize visibility, and while each path covers only a small area, together they almost cover everything. Figure 3(c) presents the expected worst-case detection probability as a function of the maximum allowed path length.

5 Related Work

The problem of computing randomized sensing strategies in adversarial environments has been studied by several groups.

Kiekintveld *et al.* [2009] develop efficient algorithms for finding Stackelberg equilibria (i.e., distributions over sensing strategies which are worst-case optimal) for large security games. In contrast to this paper, these approaches do not assume that the payoff matrices are zero-sum. However, the existing approaches cannot handle general submodular objective functions, which RSENSE can. For example, they cannot model how multiple sensors can help to observe the same intrusion. Perhaps closest in spirit is an approach by Halvorson *et al.* [2009] for computing randomized sensing strategies in multi-step hider and seeker games. Their approach does handle special cases of submodular objectives. They further consider a combinatorial number of adversarial actions (paths taken by an invader). They develop a double-oracle algorithm, which iteratively increases the strategy sets of hider and seeker by computing best responses (one of them greedy), and solving a linear program at every iteration in order to compute optimal strategies for the considered sets of actions. The algorithm terminates when the best responses are contained in the actions already considered, in which case an optimal solution has been found. However, the worst-case number of iterations required by this algorithm may be exponential. The RSENSE algorithm, in contrast, is guaranteed to obtain near-optimal solutions in a polynomial number of iterations. Furthermore, in our experiments on a problem that both algorithms can handle, we show that RSENSE outperforms the LP-based algorithm in terms of running time for large numbers of sensors.

Constrained maximization of submodular functions has been exploited for sensor placement and information gathering tasks (*c.f.*, Krause and Guestrin, 2007). However, in contrast to this work, existing approaches do not address worst-case performance. An exception is the SATURATE algorithm [Krause *et al.*, 2008], which however cannot be used to generate randomized strategies as considered in this paper. Further note that RSENSE benefits from existing work on constrained submodular maximization, which can be used as black-box subroutine for computing best responses. Streeter and Golovin [2008] develop an algorithm for the related problem of online maximization of submodular functions. They show that their algorithm achieves no $(1 - 1/e)$ regret. While their algorithm is guaranteed to produce a sequence of solutions for repeated play which performs near-optimally, it cannot handle more complex constraints (such as planning informative trajectories for mobile sensors). Kakade *et al.* [2009] develop a framework for turning general approximation algorithms into no-regret algorithms. However, they require that the objective functions are linear (modular), and hence their results do not apply in our setting.

6 Conclusions

We tackled the problem of computing randomized sensing strategies in adversarial environments. We developed the RSENSE algorithm, which applies whenever the individual objective functions associated with the intrusions are monotonic submodular, which is the case in many sensing tasks. We proved that RSENSE can efficiently obtain provably near-optimal distributions. More generally, we proved that any α-approximation algorithm for computing best responses against mixed strategies in matrix games can be used to obtain α-approximate mixed strategies. We also considered trading off worst-case and average-case performance. We developed the TRSENSE algorithm and proved that it attains a near-optimal tradeoff. We extensively evaluated and demonstrated the effectiveness of our algorithms on two real-world sensing case studies. We believe that our results provide new insights for both information gathering and solving large matrix games in general.

Acknowledgments. The authors would like to thank Matthew Streeter and Robert Schapire for helpful discussions. This research was partially supported by ONR grant N00014-09-1-1044, NSF grants CNS-0932392 and IIS-0953413, and the Caltech Center for the Mathematics of Information.

References

[Das and Kempe, 2008] A. Das and D. Kempe. Algorithms for subset selection in linear regression. In *STOC*, pages 45–54, 2008.

[Feige, 1998] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[Freund and Schapire, 1999] Y. Freund and R. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.

[Fujishige, 2005] S. Fujishige. *Submodular Functions and Optimization*. Elsevier, 2nd edition, 2005.

[Halvorson *et al.*, 2009] Erik Halvorson, Vincent Conitzer, and Ronald Parr. Multi-step multi-sensor hider-seeker games. In *IJCAI*, 2009.

[Harmon *et al.*, 2006] T. Harmon, R. Ambrose, R. Gilbert, J. Fisher, M. Stealey, and W. Kaiser. High resolution river hydraulic and water quality characterization using rapidly deployable networked infomechanical systems (nims rd). Technical Report 60, CENS, 2006.

[Kakade *et al.*, 2009] S. Kakade, A. Kalai, and K. Ligett. Playing games with approximation algorithms. *SIAM Journal of Computing*, 39(3):1088–1106, 2009.

[Kalai and Vempala, 2005] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *J. Comput. System Sci.*, 71:291–307, 2005.

[Kiekintveld *et al.*, 2009] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. In *AAMAS*, 2009.

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

[Krause *et al.*, 2008] A. Krause, B. McMahan, C. Guestrin, and A. Gupta. Robust submodular observation selection. *JMLR*, 9:2761–2801, 2008.

[Nemhauser *et al.*, 1978] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Math. Prog.*, 14(1):265–294, 1978.

[Singh *et al.*, 2009] A. Singh, A. Krause, C. Guestrin, and W. Kaiser. Efficient informative sensing using multiple robots. *JAIR*, 34:707–755, 2009.

[Streeter and Golovin, 08] M. Streeter and D. Golovin. An online algorithm for maximizing submodular functions. In *NIPS*, '08.

[Vondrák, 08] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, '08.