Nonmyopic Adaptive Informative Path Planning for Multiple Robots

Amarjeet Singh
UCLA

Andreas Krause
Caltech

William J. Kaiser
UCLA

Abstract

Many robotic path planning applications, such as search and rescue, involve uncertain environments with complex dynamics that can be only partially observed. When selecting the best subset of observation locations subject to constrained resources (such as limited time or battery capacity) it is an important problem to trade off exploration (gathering information about the environment) and exploitation (using the current knowledge about the environment most effectively) for efficiently observing these environments. Even the nonadaptive setting, where paths are planned before observations are made, is $\mathbf{NP}$-hard, and has been subject to much research.

In this paper, we present a novel approach to *adaptive* informative path planning that addresses this exploration-exploitation tradeoff. Our approach is *nonmyopic*, i.e. it plans ahead for possible observations that can be made in the future. We quantify the benefit of exploration through the "adaptivity gap" between an adaptive and a nonadaptive algorithm in terms of the uncertainty in the environment. Exploiting the submodularity (a diminishing returns property) and locality properties of the objective function, we develop an algorithm that performs provably near-optimally in settings where the adaptivity gap is small. In case of large gap, we use an objective function that simultaneously optimizes paths for exploration and exploitation. We also provide an algorithm to extend any single robot algorithm for adaptive informative path planning to the multi robot setting while approximately preserving the theoretical guarantee of the single robot algorithm. We extensively evaluate our approach on a *search and rescue* domain and a scientific monitoring problem using a real robotic system.

1 Introduction

Many robotic path planning applications involve uncertain environments with complex dynamics. For example, search and rescue in a disaster struck environment involves finding and rescuing the survivors with unknown locations using noisy sensors. With limited number of resource-constrained (limited time or battery capacity) robots available for monitoring such environments, it is essential to perform path planning to visit a subset of locations that are most "informative".

Informative path planning – selecting the best locations to observe subject to given sensing constraints, in such uncertain environments necessitates a trade off between *exploration* (gathering information about the environment) and *exploitation* (using the current belief about the state of the environment most effectively). We distinguish two different classes of algorithms: Nonadaptive (offline) algorithms, that plan and commit to the paths before any observations are made, and adaptive (online) algorithms, that update and replan as new information is collected. Most of the previous work has either dealt with approximation algorithms for the nonadaptive setting (that also is a challenging, $\mathbf{NP}$-hard optimization problem) [Singh *et al.*, 2007; Meliou *et al.*, 2007; Hollinger and Singh, 2008] or proposed (often myopic, i.e., limited look-ahead) heuristics for the adaptive setting that do not have any approximation guarantees [Stachniss *et al.*, 2005]. Partially Observable Markov Decision Processes (POMDPs) have been used to perform adaptive path planning in complex environments [Roy *et al.*, 2005]. However, such algorithms are only capable of near-optimally solving small instances of the problem.

In this paper we first propose a novel near-optimal algorithm for nonadaptive informative path planning which empirically outperforms existing nonadaptive algorithms. We then extend this nonadaptive algorithm to the adaptive setting. Our NAIVE- Nonmyopic, Adaptive, InformatiVE path planning algorithm, performs sequential model update and uses the nonadaptive algorithm as subroutine for replanning. We explicitly analyze the "adaptivity gap" (the potential performance gain of adaptive vs. nonadaptive algorithms), which allows us to give strong theoretical bounds on the suboptimality of our NAIVE algorithm. The previous state of the art in nonadaptive informative path planning is based on optimization of submodular set functions [Singh *et al.*, 2007], which model a diminishing returns property that naturally arises in many information gathering tasks [Krause *et al.*, 2006, 2008]. Our analysis is based on a novel extension of submodular set functions to adaptive sensing policies. For problems with a large adaptivity gap, we present an algorithm based on a multi-criterion utility function that seeks to simultaneously optimize exploitation (e.g., to rescue a large number of survivors) and exploration (e.g., to localize the survivors).

We also provide a generic *sequential-allocation* algorithm to extend any adaptive single robot path planning algorithm to multi robot optimization while (almost) preserving the approximation guarantee of single robot path planning algorithm. Specifically, the primary contributions of our work are:

- A new algorithm – pSPIEL$_{OR}$ for near-optimal non-adaptive informative path planning for a single robot.
- A general algorithm – NAIVE– for single robot *adaptive* informative planning. By exploiting the fact that we can bound the "adaptivity gap" for interesting real world problems (demonstrated here for a search and rescue application and a scientific monitoring task), we can prove that the adaptive solutions obtained using NAIVE, using any near-optimal non-adaptive algorithm (such as pSPIEL$_{OR}$) as a subroutine, are provable competitive with the $\mathbf{NP}$-hard optimal adaptive solution.
- A *sequential-allocation* algorithm for efficiently and near-optimally extending any algorithm for planning informative adaptive policies for a single robot (such as NAIVE or a POMDP solver) to the multi robot setting.
- The extensive evaluation of our proposed algorithms for two case studies, including a search and rescue problem and scientific monitoring using a real robotic system.

2 Problem statement

We are given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a discrete set (of locations) $\mathcal{V}$ connected by edges $\mathcal{E}$, wherein the set $\mathcal{V}$ is simultaneously traversed by k mobile robots. To model the information gathering, we associate a random variable $\mathcal{X}_s$ with each location $s \in \mathcal{V}$. The random vector $\mathcal{X}_\mathcal{V} = (\mathcal{X}_1, \ldots, \mathcal{X}_n)$ describes the (unobserved) state of the world. In our search and rescue example, $\mathcal{X}_\mathcal{V}$ models the distribution of survivors in a disaster struck environment. We assume that we have a prior joint distribution $P(\mathcal{X}_\mathcal{V})$ that models the dependencies among the random variables across space. We discuss details about these models for both our search and rescue and scientific monitoring examples in Section 6.

When a robot i visits a location $s \in \mathcal{V}$, it partially observes the state $\mathcal{X}_\mathcal{V}$ by making a noisy observation $\mathcal{Y}_s = y_s$, according to some distribution $P(\mathcal{Y}_s \mid \mathcal{X}_\mathcal{V})$. In the search and rescue example, the random variables $\mathcal{Y}_s$ model possible false positives and negatives of the robots' noisy sensors (*c.f.,* Section 6). In the adaptive setting, after making the observations, the robots then update the belief about the state of world, $P(\mathcal{X}_\mathcal{V} = \mathbf{x}_\mathcal{V} \mid \mathcal{Y}_s = y_s)$ and use the updated belief to choose the next observation location adjacent to s.

To quantify "informativeness" we define a function $u(\mathcal{A}, \mathbf{x}_\mathcal{V})$ that measures the utility of observing a set $\mathcal{A} \subseteq \mathcal{V}$ of locations if the world is in state $\mathbf{x}_\mathcal{V}$. The robots choose their movement to cooperatively maximize this utility. In the adaptive setting, the robots can select a different sequence $\pi(\mathbf{x}_\mathcal{V}) = (s_1, \ldots, s_B)$ of locations for each possible belief about the state of the world. Note that, depending on the context, we use the notation $\pi(\mathbf{x}_\mathcal{V})$ to refer both to the *sequence* $(s_1, \ldots, s_B)$ and the *set* $\{s_1, \ldots, s_B\}$ of locations visited. We call the location selection function π a *policy*. We only restrict policies to necessarily be *sequential* (i.e., a location s_t in a sequence $(s_1, \ldots, s_B)$ is chosen only based on observations $y_{t'}$ where $t' < t$), and respect the graph $\mathcal{G}$ (i.e., there must exist an edge between each s_t and s_{t+1} in $\mathcal{E}$).

For each possible policy π we compute its expected utility as $U(\pi) = \int P(\mathbf{x}_\mathcal{V}) u(\pi(\mathbf{x}_\mathcal{V}), \mathbf{x}_\mathcal{V}) d\mathbf{x}_\mathcal{V}$. In the search and rescue domain, $U(\pi)$ denotes the expected number of survivors rescued when observations are made as per the policy π. Our goal, then, is to find a policy π that maximizes $U(\pi)$. We call a policy *nonadaptive* if it selects the same sequence of locations independently of the state of the world.

For each sequence $\mathcal{A}$ of locations we also associate a cost $C(\mathcal{A})$ that corresponds to the total number of time steps required to visit all locations in $\mathcal{A}$. The total cost of a policy π is then given as $C(\pi) = \int P(\mathbf{x}_\mathcal{V}) C(\pi(\mathbf{x}_\mathcal{V})) d\mathbf{x}_\mathcal{V}$. We also associate a (possibly different) starting ($s_1^{(i)}$) and finishing location ($s_B^{(i)}$) with each robot. We call Π_i the set of all feasible policies starting at $s_1^{(i)}$ and finishing at $s_B^{(i)}$, with $C(\pi)$ upper bounded by some given budget B. Based on this notation, our goal is to solve the following optimization problem:

$$\max U(\pi_1 \cup \cdots \cup \pi_k) \text{ s.t. } \pi_i \in \Pi_i \text{ for all } i, \qquad (1)$$

where $\pi_1 \cup \ldots \cup \pi_k$ is the set of locations selected by all robots.

3 Structure in informative path planning

As shown by Krause *et al.* [2006], many problems of non adaptively choosing locations satisfy the following intuitive diminishing returns property: Adding a new location $s \in \mathcal{V}$ helps more if we have selected only a few locations so far, than if we have already selected many locations. This intuition is formalized by the concept of *submodular set functions*: A function $F : 2^\mathcal{V} \rightarrow \mathcal{A}$ is called *submodular* [Nemhauser *et al.*, 1978] if whenever $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and $s \in \mathcal{V} \setminus \mathcal{B}$ it holds that $F(\mathcal{A} \cup \{s\}) - F(\mathcal{A}) \geq F(\mathcal{B} \cup \{s\}) - F(\mathcal{B})$.

In this paper, we generalize this notion to *policies*. For two policies π and π', we say $\pi \subseteq \pi'$ if $\pi(\mathbf{x}_\mathcal{V}) \subseteq \pi'(\mathbf{x}_\mathcal{V})$ $\forall \mathbf{x}_\mathcal{V}$, i.e., in any state of the world, π selects a subset of observation locations selected by π'. We say $|\pi| \leq m$ if $|\pi(\mathbf{x}_\mathcal{V})| \leq m$ $\forall \mathbf{x}_\mathcal{V} \in \mathcal{X}_\mathcal{V}$. Similarly, we generalize other set operations $\pi_1 \cup \pi_2$, $\pi_1 \cap \pi_2$, $\pi_1 \setminus \pi_2$ etc. A function U that maps *policies* to real numbers is called submodular if for all policies π_1, π_2, π' s.t. $\pi_1 \subseteq \pi_2$, it holds that $U(\pi_1 \cup \pi') - U(\pi_1) \geq U(\pi_2 \cup \pi') - U(\pi_2)$, i.e., adding π' to π_1 will lead to bigger improvement in expected utility than adding π' to π_2. Another intuitive property is that the utility function U is monotonic, i.e., as more locations are selected, the expected utility never decreases. Formally, a utility function over policies is called monotonic if $U(\pi_1) \leq U(\pi_2)$ whenever $\pi_1 \subseteq \pi_2 \subseteq \mathcal{V}$.

In addition to *submodularity* and monotonicity, several spatial information gathering problems, including search and rescue and environmental monitoring, exhibit another important *locality* property: Observation locations which are very far apart are approximately independent. This implies that if we observe a subset of locations $\mathcal{A}_1$ in one region, and a set $\mathcal{A}_2$ in another region (far apart), then $U(\mathcal{A}_1 \cup \mathcal{A}_2) \approx U(\mathcal{A}_1) + U(\mathcal{A}_2)$. This property can be abstracted by assuming that there are constants $r > 0$ and $0 < \gamma \leq 1$, such that for any sets $\mathcal{A}_1$ and $\mathcal{A}_2$ which are at least distance r apart, $U(\mathcal{A}_1 \cup \mathcal{A}_2) \geq U(\mathcal{A}_1) + \gamma U(\mathcal{A}_2)$. Such set functions U are called

(r,γ)-*local* [Krause *et al.*, 2006]. In this paper, we generalize this notion to policies: We say U is (r,γ)-local if for any policies π_1 and π_2, s.t. the sets $\pi_1(\mathbf{x}_\mathcal{V})$ and $\pi_2(\mathbf{x}_\mathcal{V})$ are distance r apart in any state $\mathbf{x}_\mathcal{V}$, then $U(\pi_1 \cup \pi_2) \geq U(\pi_1) + \gamma U(\pi_2)$.

One way to construct (r,γ)-local, submodular, monotonic functions on policies is given by the following proposition:

Proposition 1. *If $u(\mathcal{A}, \mathbf{x}_\mathcal{V})$ is (r,γ)-local, submodular and monotonic in $\mathcal{A}\ \forall\ \mathbf{x}_\mathcal{V}$, then $U(\pi) = \int P(\mathbf{x}_\mathcal{V}) u(\pi(\mathbf{x}_\mathcal{V}), \mathbf{x}_\mathcal{V})$ is (r,γ)-local, submodular and monotonic.*

All the proofs for this paper are discussed in the Appendix. Below, we will only assume that the expected utility function U is (r,γ)-local, submodular and monotonic.

4 Single robot informative path planning

In this section, we develop NAIVE, a *N*onmyopic *A*daptive *I*nformati*VE* path planning algorithm for a single robot ($k = 1$). We extend the algorithm to multiple robots in Section 5. Our algorithm is based on the following approach. We start with a prior belief $P(\mathcal{X}_\mathcal{V})$ about the state of the world. Based on this belief, we nonmyopically plan a nonadaptive path $\mathcal{A}$ maximizing the expected utility $U(\mathcal{A})$ as per our current belief. The robot then moves to the first location s on the selected path, makes the observation $\mathcal{Y}_s = y_s$, updates its belief $P(\mathcal{X}_\mathcal{V} \mid y_s)$, and replans according to its posterior belief. In Section 4.1, we present the nonadaptive algorithm used as a subroutine for the adaptive procedure. In Section 4.2 we provide details on the adaptive procedure, and analyze its performance.

4.1 Nonadaptive informative path planning

We now present an algorithm for planning a nonadaptive policy $\pi_\mathcal{A}$ where the sequence of observation locations $\mathcal{A} = (s_1, \ldots, s_B)$ is selected by the policy $\pi_\mathcal{A}$ *independently* of the state $\mathbf{x}_\mathcal{V}$. It can be seen that if U is a (r,γ)-local submodular function on policies, then $F(\mathcal{A}) = U(\pi_\mathcal{A})$ is a (r,γ)-local submodular set function. Hence, in the nonadaptive setting we need to find a set $\mathcal{A}^*$ satisfying

$$\mathcal{A}^* = \operatorname{argmax} F(\mathcal{A}), \text{ s.t. } C(\mathcal{A}) \leq B,$$

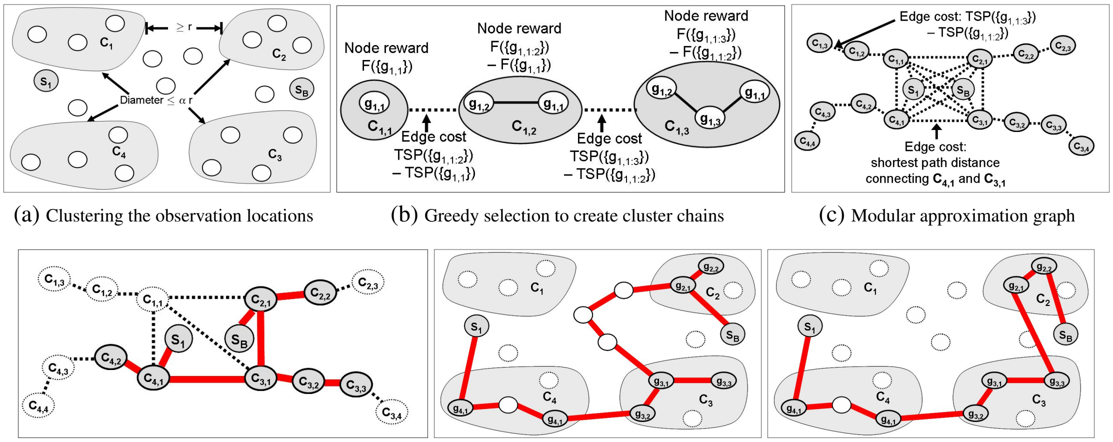
where $C(\mathcal{A})$ is the cost of the shortest path connecting the selected locations $\mathcal{A}$, starting with s_1 and finishing at s_B.

This optimization problem, seeking to maximize a submodular utility function with an upper bound on the total cost, is called *submodular orienteering problem* (introduced by Chekuri and Pal [2005], who developed a theoretical, superpolynomial algorithm). In this paper, we propose a new, efficient algorithm, *pSPIEL-Orienteering* (pSPIEL$_{OR}$), for solving the submodular orienteering problem while exploiting the local-submodular property of the utility function. The complete algorithm is illustrated in Figure 1.

1. The algorithm takes as input a starting location (s_1), a finishing location (s_B), an upper bound on the path cost B and a local-submodular set function F, conditioned on previous observations. We, first, randomly partition the locations $\mathcal{V} \setminus \{s_1 \cup s_B\}$ into *small* clusters of diameter αr (where α is a parameter). Subsequently, nodes close to the "boundary" of their clusters are stripped away, such that the remaining clusters are "well-separated" (have distance $\geq r$). (It was proved by Krause *et al.* [2006] that not too many nodes get stripped away). Due to the locality property of the utility function U, each cluster is approximately independent. This well-separated clustering is called *padded decomposition* [Gupta *et al.*, 2003], and is illustrated in Figure 1a.
2. Within the locations $\mathcal{C}_i$ of each cluster i, we then use a greedy algorithm to get an ordering $g_{i,1}, g_{i,2}, \ldots g_{i,n_i}$ on the n_i nodes (*c.f.,* Figure 1b). Hereby, $g_{i,\ell+1} = \operatorname{argmax}_{v \in \mathcal{C}_i \setminus \mathcal{G}_{i,\ell}} F(\{v\} \cup \mathcal{G}_{i,\ell})$, where $\mathcal{G}_{i,\ell} = \{s_1, s_B, g_{i,1}, \ldots, g_{i,\ell}\}$. These nodes are then connected to form a chain for the cluster using cost for the edge $(g_{i,j}, g_{i,j+1})$ to be the minimum cost required to reach the node $j+1$ from the first j nodes already selected in the chain. This minimum cost is calculated by adding the cost of edges (as per $\mathcal{E}$) in the shortest path traversed to reach the node $j+1$ from the first j nodes. With each node $g_{i,j}$, we associate an additive reward $r_{i,j} = F(\mathcal{G}_{i,j}) - F(\mathcal{G}_{i,j-1})$. The submodularity of F ensures that the first k nodes in this chain are almost as informative as the best subset of k nodes in the cluster [Guestrin *et al.*, 2005].
3. Next, we create a "modular approximation graph" $\mathcal{G}'$ from $\mathcal{G}$ by taking all these chains, and creating a fully connected graph on $s_1, s_B, g_{1,1}, g_{2,1}, \ldots, g_{m,1}$, the starting and finishing nodes and all the first nodes of each chain. The edge costs are represented by the shortest path distances between the corresponding nodes and calculated by adding the cost of the edges (as per $\mathcal{E}$) traversed in this shortest path (*c.f.,* Figure 1c).
4. Constrained by budget B, we then use an existing *modular orienteering* algorithm [Chekuri *et al.*, 2008] on $\mathcal{G}'$ with s_1 and s_B as starting and finishing node respectively (*c.f.,* Figure 1d). Note that modular orienteering is a special case of submodular orienteering where all observations are fully independent, i.e., utility function $u(\cdot, \mathbf{x}_\mathcal{V})$ is additive.
5. The selected path in $\mathcal{G}'$ is then expanded in terms of the corresponding shortest path in $\mathcal{G}$ (*c.f.,* Figure 1e). Since each edge in $\mathcal{G}'$ may represent multiple edges spanning through locations in $\mathcal{G}$, the corresponding expanded path in $\mathcal{G}$ will now (possibly) have additional nodes from $\mathcal{G}$ that were absent in $\mathcal{G}'$.
6. Finally, *tour-opt* heuristics by Lin [1965] are applied to smooth out and shorten the path over the selected locations (*c.f.,* Figure 1f).

The pSPIEL$_{OR}$ algorithm and its analysis is based on the pSPIEL algorithm, originally proposed for the purpose of communication-efficient informative sensor placement, by Krause *et al.* [2006]. However, it implements necessary modifications for path planning such as handling a starting (s_1) and finishing (s_B) location and applying the modular orienteering algorithm followed by path expansion and smoothening in steps 4-6. We prove the following result about pSPIEL$_{OR}$:

Proposition 2. pSPIEL$_{OR}$ *finds a path $\mathcal{A}$ with cost $\mathcal{O}(r \dim(\mathcal{V}, E)) \times B$ and expected $F(\mathcal{A}) \geq \Omega(\gamma) \times F(\mathcal{A}^*)$.*



(a) Clustering the observation locations (b) Greedy selection to create cluster chains (c) Modular approximation graph

(d) Perform modular orienteering (e) Path expansion in original $\mathcal{G}$ (f) Path smoothening

Figure 1: Illustration of nonadaptive informative path planning using pSPIEL$_{OR}$ (*pSPIEL-Orienteering*) algorithm.

where $\mathcal{A}^*$ is the optimal set chosen for this problem and $\dim(\mathcal{V}, E)$ is the *doubling dimension*, which is constant for many graphs and is $\mathcal{O}(\log n)$ for arbitrary graphs [Gupta *et al.*, 2003]. Running time of pSPIEL$_{OR}$ is polynomial in number of observation locations. In Section 6, we empirically show that pSPIEL$_{OR}$ outperforms state of the art algorithms for submodular orienteering for the search and rescue problem.

4.2 Adaptive informative path planning

We now describe NAIVE– our *N*onmyopic *A*daptive *I*nformati*VE* path planning algorithm, based on an iterative Bayesian updating and replanning approach. The algorithm is initialized with specified starting (s_1) and ending locations (s_B) and an upper bound of B timesteps on the path cost. In first timestep, NAIVE applies a nonadaptive algorithm, such as pSPIEL$_{OR}$, using the utility function $F(\mathcal{A} \mid y_s) = \int P(\mathbf{x}_\mathcal{V} \mid \mathbf{y}_s) u(\mathcal{A}, \mathbf{x}_\mathcal{V}) d\mathbf{x}_\mathcal{V}$ representing the "conditional utility" conditioned on observation $\mathcal{Y}_s = \mathbf{y}_s$ made at s_1. The algorithm then moves the robot to the next location on the selected nonadaptive path and iteratively use pSPIEL$_{OR}$ with updated starting location s_1 and budget $B-1$, while keeping the finishing location fixed at s_B, in the subsequent timesteps. Algorithm 1 outlines this non-myopic, adaptive algorithm.

Let π_{NAIVE} be the location selection policy induced by the NAIVE algorithm. It can be seen that π_{NAIVE} performs at least as well as the nonadaptive path $\pi_\mathcal{A}$ that pSPIEL$_{OR}$ returns, i.e., $U(\pi_{NAIVE}) \geq U(\pi_\mathcal{A})$. Moreover, we have the following result that compares the performance of NAIVE to the performance of an optimal adaptive policy.

Theorem 3. *Suppose*

$$\Gamma = \mathbb{E}_{\mathbf{y}_\mathcal{V}}[\max_{|\mathcal{A}| \leq B} F(\mathcal{A} \mid \mathbf{y}_\mathcal{V})] / \max_{|\mathcal{A}| \leq B} \mathbb{E}_{\mathbf{y}_\mathcal{V}}[F(\mathcal{A} \mid \mathbf{y}_\mathcal{V})].$$

Then

$$\mathbb{E}[U(\pi_{NAIVE})] \geq \Omega(\gamma/\Gamma) U(\pi^*),$$

where π^* is an optimal sequential policy of length $\Omega(B/(r \dim(\mathcal{V}, E)))$.

Algorithm: NAIVE-pSPIEL$_{OR}$
Input: s_1, s_B, u, B
Output: Sequence of selected locations $\pi(\mathbf{x}_\mathcal{V})$
begin
  $s \leftarrow s_1$; $\pi[1] = s$; $B_{pp} \leftarrow B$; $obs \leftarrow \{\}$;
  for $1 \leq t \leq B$ **do**
    $y_s \leftarrow observe(s)$; $obs \leftarrow obs \cup \{\mathcal{Y}_s = y_s\}$;
    $\mathcal{P}_t \leftarrow pSPIEL_{OR}(s, s_B, B_{pp}, F(\cdot \mid obs))$;
    $s \leftarrow \mathcal{P}_t[2]$; $\pi[t+1] \leftarrow s$; $B_{pp} \leftarrow B_{pp} - 1$;
  return π;
end

Algorithm 1: NAIVE-Nonmyopic Adaptive InformatiVE path planning algorithm using pSPIEL$_{OR}$.

Hereby, Γ is a (computable) upper bound to the *adaptivity gap*, a fundamental quantity in active learning, defined as the ratio

$$\big(\max_{|\pi| \leq B} U(\pi)\big) / \big(\max_{|\mathcal{A}| \leq B} U(\pi_\mathcal{A})\big) \leq \Gamma,$$

i.e., the ratio of the performance of the optimal policy divided by the performance of the best nonadaptive path. We explicitly bound Γ for the two case studies of search and rescue and hotspot sampling in Section 6.

Theorem 3, along with the fact that we can bound the adaptivity gap for interesting real-world applications (Propositions 5 and 6) implies that the adaptive solutions obtained by the NAIVE algorithm, using any near-optimal non-adaptive algorithm (such as pSPIEL$_{OR}$ or eMIP) as a subroutine, are provably competitive with the **NP**-hard optimal adaptive solution.

4.3 Trading off exploration and exploitation

If the adaptivity gap Γ is small, Theorem 3 proves near-optimal performance of the NAIVE algorithm. In the case of large adaptivity gap, we choose the following algorithm to select paths that simultaneously exploit the current belief about the state of the world as well as decrease the bound on the adaptivity gap as much as possible. We define two utility functions, U_1 and U_2. The second utility, $U_2(\pi) = \int P(\mathbf{x}_\mathcal{V}) u(\pi(\mathbf{x}_\mathcal{V}), \mathbf{x}_\mathcal{V}) d\mathbf{x}_\mathcal{V}$, seeks to exploit our current belief of the world. The first utility U_1 seeks to make observations that are likely to reduce the adaptivity gap, i.e., it is chosen such that $U_1 \to 0$ implies $\Gamma \to 0$. Both objective functions are application specific, and we will explain our choices in Section 6. We then use *scalarization* [Boyd and Vandenberghe, 2004] by setting $U(\pi) = \lambda U_1(\pi) + (1-\lambda) U_2(\pi)$, where $0 \leq \lambda \leq 1$ and $\pi(\mathbf{x}_\mathcal{V}) \in \mathcal{V}$. λ is a knob we can turn: if $\lambda = 0$, we *exploit* our current belief about the survivor locations, by seeking to observe the locations where survivors are most likely located. If $\lambda = 1$, we *explore* in order to reduce the uncertainty about the survivor locations. Values $0 < \lambda < 1$ encourage a tradeoff of exploration and exploitation. A similar analysis was used for Gaussian Processes by Krause and Guestrin [2007] (who did not consider path planning).

5 Multi robot informative path planning

One approach for extending any single robot planning algorithm to plan simultaneous paths with multiple robots is to form a new graph where each node represents the vector of locations of all k robots, and then apply the single robot algorithm to this product graph. Unfortunately, the size of this product graph grows exponentially in k, which is infeasible for large teams of robots. We now present an algorithm for multiple robot adaptive path planning that scales *linearly* in the number k of robots, while providing almost the same performance as applying the single robot algorithm to the product graph. The algorithm, *sequential allocation*, extends *any* single robot algorithm for adaptive informative path planning, such as NAIVE-pSPIEL$_{OR}$, or a POMDP solver, to the multi robot setting. Sequential allocation is a greedy algorithm, that optimizes the policy π_i for robot i conditioned on all policies $\pi_1, \ldots, \pi_{i-1}$ that have already been selected, by optimizing $\pi_i = \operatorname{argmax}_\pi U_{\pi^{(1:i-1)}}(\pi)$, where $U_\pi(\pi') = U(\pi \cup \pi') - U(\pi)$ and $\pi^{(1:i-1)} = \pi_1 \cup \cdots \cup \pi_{i-1}$. This algorithm is similar to the algorithm proposed by Singh *et al.* [2007] for the nonadaptive setting, and generalizes it to the framework involving policies. We prove:

Theorem 4. *Suppose we have an approximation algorithm that, given policies $\pi_j \in \Pi_j$ and $\pi^{(1:i)} = \pi_1 \cup \cdots \cup \pi_i$, finds a solution π_G such that $U_{\pi^{(1:i)}}(\pi_G) \geq \frac{1}{\eta} \max_{\pi \in \Pi_{i+1}} U_{\pi^{(1:i)}}(\pi)$. Then* sequential allocation *using this algorithm guarantees a solution $(\pi'_i)_i$ such that*

$$U(\pi'_1 \cup \cdots \cup \pi'_k) \geq \frac{1}{\eta + 1} \max_{\pi_1 \in \Pi_1, \ldots, \pi_k \in \Pi_k} U(\pi_1 \cup \cdots \cup \pi_k).$$

When using NAIVE-pSPIEL$_{OR}$, $\eta = \mathcal{O}(\Gamma/\gamma)$. The approximation guarantee provided by *sequential-allocation* algorithm holds even when the planning is done sequentially at each timestep, i.e., the information about the observation locations selected so far, for each robot, is known to all the planners while they select their next observation location. Further, this information may be communicated to all the planners for distributed path planning or the paths may be centrally planned by a single planner.

6 Experiments

We now present empirical results on two case studies.

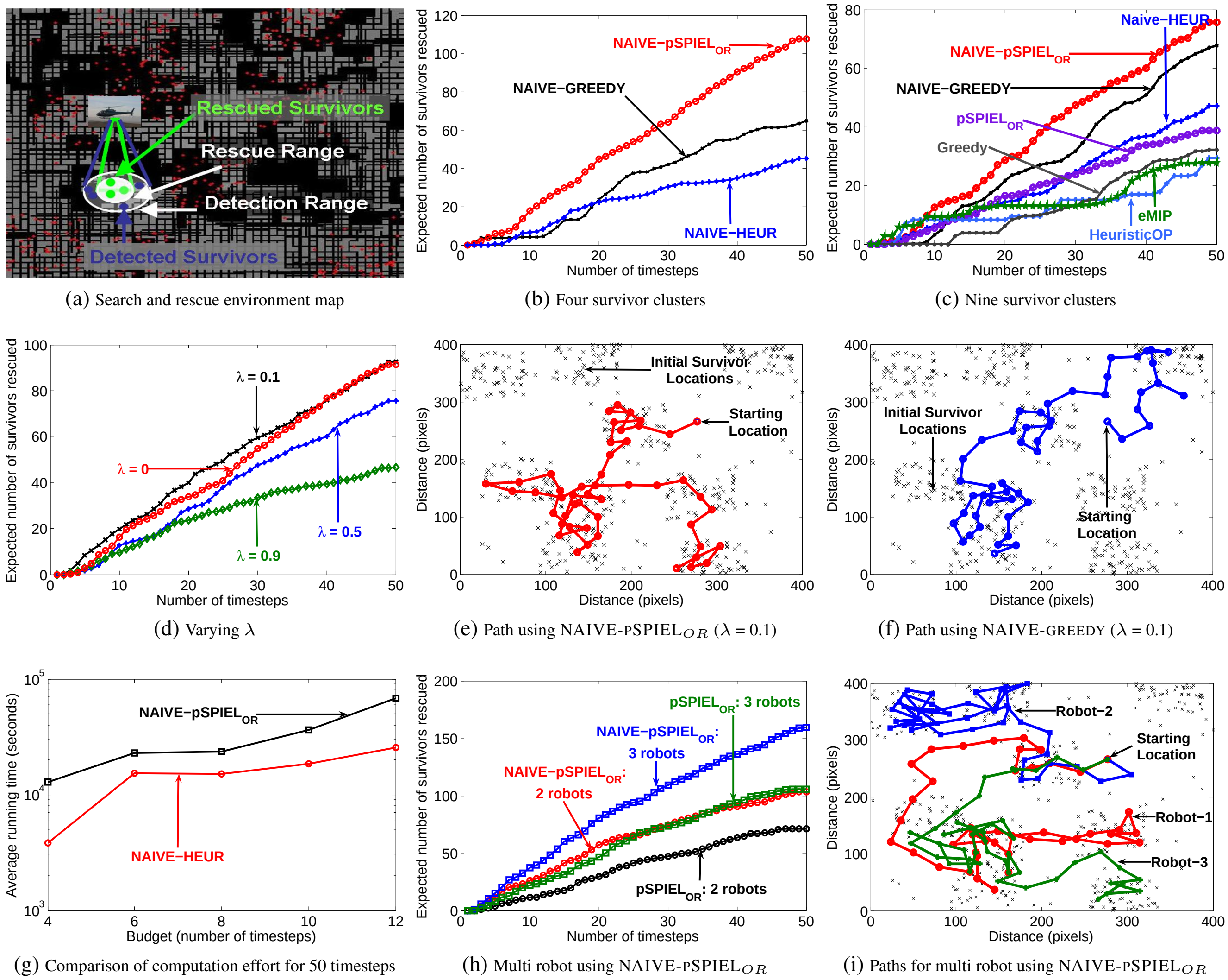
6.1 Case Study I: Search and rescue

We perform our first set of experiments on a realistic simulation environment for a search and rescue domain.

Experimental setup: This case study is based on the *Sensor Planning Research Challenge* (SPRC), originally announced for the International Conference on Machine Learning, 2007, and extended by slight modifications. The SPRC problem considers an earthquake in a large urban environment with survivors spread across heavily populated areas of interest, accessible only using rescue helicopters (*c.f.,* SPRC 2007 for details on the problem specification). Each helicopter is equipped with two capabilities – a long range, noisy sensor to detect the survivors; and the ability of rescuing the survivors using a short range sensor. Field of view of these helicopters is assumed to be occluded by the buildings and only the unobstructed survivors are assumed to be detected or rescued. Noisy detection of survivors' locations are received over a cellular network as well (signals from only a few of them are received probabilistically during each timestep).

With unknown survivor locations and noisy observations, we therefore need to tradeoff the detection (exploration) using the long range sensor and rescuing survivors at known locations (exploitation). For path planning, we discretized the search space into 1000 locations distributed uniformly at random. Cost for traversing between each of these locations is calculated in terms of number of timesteps required for the shortest path between the two locations (calculated based on robot speed - assumed to be $20 m/s$, pixel length - $5 meters$ and duration of each timestep - assumed to be $10 seconds$). We assume a total of 500 survivors distributed according to a mixture of Gaussians with (unknown) 4 (resp. 9) centers. The prior joint distribution $P(\mathcal{X}_\mathcal{V})$ for survivor locations is assumed to be uniform. A certain (unknown) number of survivors are also assumed to be mobile and can move in any direction with equal probability. Figure 2a illustrates a snapshot of one timestep during the path planning. The background represents a grayscale bitmap image with 400x400 pixels. Each pixel represents an urban region of 5 meters by 5 meters. Grayscale value of the pixel represent the height of buildings at that location. Only the survivors on ground (white region) are assumed to be not occluded in field of view and can be detected or rescued with sensors attached to the mobile robot. Red points in the figure represent locations of survivors in the beginning (uniformly distributed in 9 gaussian clusters). Green points are the (rescued) survivors within the rescue range of the mobile robot and blue points are the detected survivors within the detection range of the mobile robot.

Model: At each time instant, we receive a new set of partial observations from both the cellular network and sensors attached to the mobile robot. *Detection* information received over the cellular network is independent of the location of the mobile robot and has a noise of 5 meters associated with it. During each timestep, the survivors transmitting over the cellular network during the previous timestep stop



(a) Search and rescue environment map (b) Four survivor clusters (c) Nine survivor clusters

(d) Varying λ (e) Path using NAIVE-pSPIEL$_{OR}$ ($\lambda = 0.1$) (f) Path using NAIVE-GREEDY ($\lambda = 0.1$)

(g) Comparison of computation effort for 50 timesteps (h) Multi robot using NAIVE-pSPIEL$_{OR}$ (i) Paths for multi robot using NAIVE-pSPIEL$_{OR}$

Figure 2: Search and rescue experiments comparing the performance of NAIVE-GREEDY, NAIVE-HEUR and NAIVE-pSPIEL$_{OR}$.

transmitting with a probability of 0.1 while survivors not transmitting during the previous timestep start transmitting with a probability of 0.01.

For each helicopter location s, the attached sensors are directed vertically down. With each sensor, we associate a circular field of view for, $\mathcal{W}_{1,s} \subseteq \mathcal{V}$ and $\mathcal{W}_{2,s} \subseteq \mathcal{V}$, that provides a subset of unobstructed locations in the corresponding range (with $\mathcal{W}_{2,s} \subseteq \mathcal{W}_{1,s}$). Range for $\mathcal{W}_{1,s}$ is assumed to be 25 meters and the range for $\mathcal{W}_{2,s}$ is assumed to be 10 meters. For each $s' \in \mathcal{W}_{i,s}$, $\mathcal{X}_{s'}$ is revealed with certain false positive/negative probability p_i ($p_1 = 0.5; p_2 = 0$). The observation $\mathcal{Y}_s$ therefore is the noisy measurement of the random vector $\mathcal{X}_{\mathcal{W}_{1,s}}$ and the noise-free observation of $\mathcal{X}_{\mathcal{W}_{2,s}}$. which involves multi-person tracking and data association (details in the longer version of this paper).

Using the observations y_s, and detections using the cellular network, we then perform Bayesian inference to compute the probability density of survivor locations, $P(\mathcal{X}_s \mid y_s)$ as follows. Detections using the cellular network are assumed to be *labeled*, i.e., are always attributed to a specific survivor while the detections using the sensor attached with the mobile robot are *unlabeled*. We track each person, for which we have at least one cellular observation, using a mixture of Gaussian distribution, with one component representing the distribution over locations in case the person is static, and the other component representing the location in case the person is mobile. Variance of the "mobile" component grows with time. Additionally, we maintain an "anonymous" distribution over the location of the remaining survivors for which no cellular detection is made so far. Instead of modeling the location of each person separately, this distribution models the probability of occupancy at every grid cell.

In order to incorporate detections from the sensor attached with the mobile robots, we perform maximum likelihood data association (i.e., attribute each observation to the most likely survivor). For the Gaussian component, we perform updates by linearizing the observations. For the "anonymous" distribution, we apply a drift process to accommodate mobile targets, and track the occupancy probability at each cell using an occupancy grid style algorithm Thrun [2003]. The survivors rescued in each timestep are removed from the simulation.

The exploration utility function is the expected value of $u_1(\mathcal{A}, \mathbf{x}_\mathcal{V})$, the number of people detected at locations $\mathcal{A}$ in state $\mathbf{x}_\mathcal{V}$. The exploitation utility function is defined as the expectation over $u_2(\mathcal{A}, \mathbf{x}_\mathcal{V})$, i.e., the number of people rescued at locations $\mathcal{A}$ in state $\mathbf{x}_\mathcal{V}$. We can show the following result:

Proposition 5. *The adaptivity gap is bounded as* $\Gamma \leq \mathbb{E}_{\mathbf{y}_\mathcal{V}} \max_s P(\mathcal{X}_s \mid \mathbf{y}_\mathcal{V})/P(\mathcal{X}_s)$.

Hence the adaptivity gap is bounded by the maximum "amplification" in the occupancy probability $P(\mathcal{X}_s)$ that can occur by knowing all observations. Note that the above bound on Γ can be approximately computed with arbitrarily small additive error by sampling possible observation vectors $\mathbf{y}_\mathcal{V}$.

Experimental results: We empirically compare the performance of NAIVE-PSPIEL$_{OR}$ with two other algorithms - (a) NAIVE-GREEDY with a greedy algorithm as subroutine in Line 1 of Algorithm 1; and (b) NAIVE-HEUR with a path planning heuristic proposed by Chao *et al.* [1996] (empirically shown to provide efficient results in the nonadaptive setting by Singh *et al.* [2007]) as subroutine. For sake of brevity, we refer to the nonadaptive greedy algorithm as $greedy$ and the nonadaptive heuristic algorithm as $heuristicOP$.

For both NAIVE-HEUR and NAIVE-PSPIEL$_{OR}$, we fixed the starting and finishing location as the current location. We fixed budget B to 50 timesteps and exploration-exploitation parameter λ to 0.5. Each experiment was run multiple times to compute the expected utility. To reduce the computation effort for the nonadaptive path planning algorithm, we fixed the lookahead to 8 timesteps. Lookahead budget was empirically found to have insignificant influence on the performance of the corresponding adaptive algorithm.

Figures 2b and 2c compare the expected utility (expected number of survivors rescued) for NAIVE-GREEDY, NAIVE-HEUR and NAIVE-PSPIEL$_{OR}$ with survivors scattered in 4 and 9 clusters respectively. For smaller number of clusters, the expected utility gap widens between nonmyopic planning using NAIVE-PSPIEL$_{OR}$ and myopic NAIVE-GREEDY and NAIVE-HEUR. Figure 2c compares the performance of all the three algorithms with their corresponding nonadaptive versions. We also compared with the performance of an efficient nonadaptive path planning algorithm (with provable approximation guarantees) proposed by Singh *et al.* [2007], referred to in the figure as EMIP. As expected, the adaptive algorithms provide better expected utility compared to the nonadaptive algorithms.

Figure 2d compares the expected utility while performing path planning using NAIVE-PSPIEL$_{OR}$ for different values of exploration-exploitation parameter, λ. For the given problem setting, the utility function favors exploitation of the current knowledge about survivor locations ($\mathcal{X}_\mathcal{V}$) in larger expected utility for smaller λ. Exploration benefit (to achieve higher expected utility) between $\lambda = 0.1$ and $\lambda = 0$ also diminishes with time. Figure 2e and Figure 2f illustrate paths for NAIVE-PSPIEL$_{OR}$ and NAIVE-GREEDY for λ = 0.1 respectively. Nonmyopic planning using NAIVE-PSPIEL$_{OR}$ results in seeking out more number of survivor clusters (even when they are at a large distance from the current location) more quickly than myopic NAIVE-GREEDY.

Figure 2g illustrates the comparison of computation effort (on a standard dual core desktop) between NAIVE-PSPIEL$_{OR}$ and NAIVE-HEUR for an experiment running over 50 timesteps with varying budget inputs. With polynomial time running time guarantee, corresponding empirical validation of low computation effort, as shown in Figure 2g and significant increase in expected number of survivors rescued (c.f. Figure 2c), NAIVE-PSPIEL$_{OR}$ can be used for efficient path planning in search and rescue domain.

Figure 2h illustrates the performance of *sequential-allocation* when using NAIVE-PSPIEL$_{OR}$ and PSPIEL$_{OR}$ as the single robot path planning algorithms. When using NAIVE-PSPIEL$_{OR}$, each robot exchanges the the collected observation information with other robots before performing path planning to decide the next observation location. Since each of the multiple robots can now (possibly) explore different regions of the environment, increasing the number of robots from two to three when using NAIVE-PSPIEL$_{OR}$ results in higher performance gain than when using PSPIEL$_{OR}$ (non adaptive path planning). Figure 2i illustrates typical paths taken by each robot when a total of 3 robots were available for path planning using *sequential-allocation*.

6.2 Case Study II: Hotspot sampling

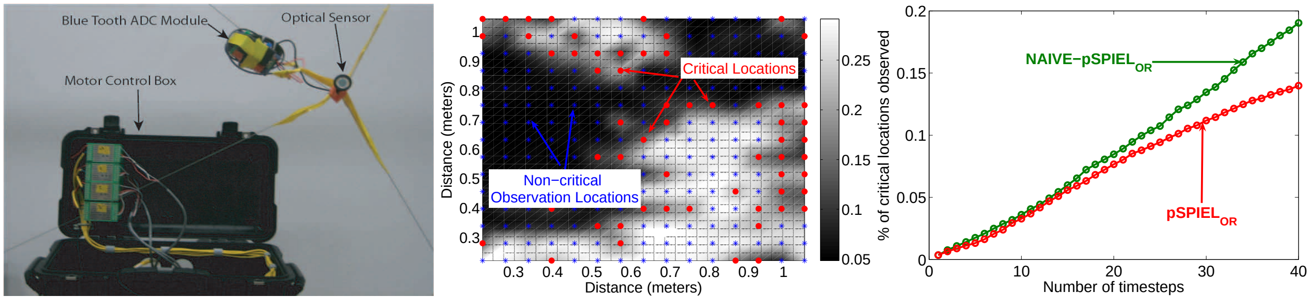
Our second case study applies informative path planning to the scientific application of monitoring photosynthetically active regions in a forest understory. Tropical forests, in particular the small plants growing in the below-canopy environments, play a significant role in global climate change [Houghton, 2005]. Spatial and temporal variation of light intensity in such environments is not well known and its measurement is technically challenging.

Experimental setup: We collected a series of 10 images, each separated by ≈ 20 minutes, from 8:30 - 11:30 AM, using a down-looking camera to capture light intensity distribution under a tree canopy at San Jacinto mountains reserve (Southern California). A camera can only capture the reflected light intensity. However, the incident light intensity is of importance for the photosynthesis process. Accurate light measurements require a physical sensing system and cannot be captured using a camera. We only used these images to model the complex dynamics of such environments. We projected these images onto a planar surface to be sampled using a light sensor attached with Planar Networked Info Mechanical System (NIMS-PL, Borgstrom *et al.* [2008], *c.f.,* Figure 3a). We discretized the image into a uniform grid of 15×15 locations for path planning.

Model: We modeled the observed light intensity as a multivariate normal distribution over the grid locations $\mathcal{X}_\mathcal{V}$:

$$P(\mathcal{X}_\mathcal{V} = \mathbf{x}_\mathcal{V}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_\mathcal{V}-\mu)^T\Sigma^{-1}(\mathbf{x}_\mathcal{V}-\mu)}.$$

In consultation with the biologists, we applied a suitable transformation to the collected intensity data to identify the



(a) NIMS-PL system (b) Test image for performance comparison (c) Performance comparison

Figure 3: Hot spot sampling using NIMS-PL comparing an offline pSPIEL$_{OR}$ with online NAIVE-pSPIEL$_{OR}$.

set of locations critical for photosynthesis (for the specific plant species growing in the observed region). The goal of this monitoring problem is observe as many critical regions as possible. Hence, we define the exploitation utility function, U_2, as expectation over $u(\mathcal{A}, \mathbf{x}_\mathcal{V})$, the number of critical locations in $\mathcal{A}$ given true intensity $\mathbf{x}_\mathcal{V}$. The exploration utility function U_1 quantifies the expected reduction in uncertainty for the multivariate gaussian $(\operatorname{tr}\Sigma_\mathcal{V} - \operatorname{tr}\Sigma_{\mathcal{V}|\mathcal{A}})$. In this case, Σ represents the prior joint distribution $P(\mathcal{X}_\mathcal{V})$, learned empirically from the remaining projected images other than the one used for performing path planning (*leave-one-out cross-validation*). After observing each location $\mathcal{Y}_s$, we perform Bayesian inference, updating our belief about the joint light distribution, conditioned on the observations, i.e., we compute posterior mean $\mu_{\mathcal{V}|\mathcal{Y}_s}$ and covariance $\Sigma_{\mathcal{V}|\mathcal{Y}_s}$. We can prove an analogous statement to Proposition 5:

Proposition 6. *The adaptivity gap is bounded as* $\Gamma \leq \mathbb{E}_{\mathbf{y}_\mathcal{V}} \max_s P([\mathcal{X}_s \in \mathcal{R}] \mid \mathbf{y}_\mathcal{V})/P(\mathcal{X}_s \in \mathcal{R})$ *where* $\mathcal{R}$ *represents the "interesting range" of light intensity.*

Hence the adaptivity gap is bounded by the maximum increase in the expected number of "photosynthetically interesting" observation locations that can occur by knowing all observations. Red dots in Figure 3b represent all the photosynthetically interesting locations in the corresponding image.

Experimental results: We fixed λ to 0.5, and path planning budget (B_{pp}) for nonadaptive path planning algorithm to 10 timesteps (to reduce the computation effort) and budget (B) to 40. Figure 3c compares the expected percentage of critical observation locations observed in each of the projected images using NAIVE-pSPIEL$_{OR}$ and pSPIEL$_{OR}$. The X-axis represent the number of timesteps. As the time progresses, the average utility gap between an online and an offline algorithm increases. This result demonstrates that adaptive path planning can successfully take into account observations to predict relevant regions for further monitoring.

7 Conclusions

In this paper we presented a nonmyopic algorithm for informative path planning using multiple robots. Our adaptive algorithm – NAIVE– works by applying a novel nonadaptive path planning algorithm – pSPIEL$_{OR}$– as a subroutine at each timestep with the updated belief about the state of the environment to decide the next observation location. We analyze the performance of NAIVE using the concept of submodular functions, which we generalize to sequential policies. We bound the algorithm's suboptimality by analyzing the adaptivity gap, that quantifies the benefit of the optimal adaptive over the optimal nonadaptive policy.

For the case of a small adaptivity gap, our algorithm exhibits provably near-optimal performance guarantees for efficiently monitoring the complex uncertain environments, outperforming state of the art nonadaptive algorithms. To handle large adaptivity gaps, we use a multicriterion utility function that seeks to simultaneously optimize exploitation of the current belief about the state of the environment and exploration to decrease the uncertainty about state of the environment.

We also provide a generic *sequential-allocation* algorithm for extending *any* single robot path planning algorithm in this setting, such as NAIVE-pSPIEL$_{OR}$, for multi robot optimization while (almost) preserving the approximation guarantee of the single robot path planning algorithm. We extensively evaluate our approach on the search and rescue domain and on an actual robotic system for the critical application of monitoring light intensity in the forest understory. We believe that our results can benefit the development and analysis of new adaptive algorithms and their application to complex scientific and societal problems.

Acknowledgements This work was supported by NSF Grant Nos. ECCS-0725441 and CNS-0331481. We would also like to thank Zainul Charbiwala for his comments to improve the paper presentation. Amarjeet Singh was partially supported by UCLA Dissertation Year Fellowship. Andreas Krause was partially supported by a Microsoft Research Graduate Fellowship.

References

P. H. Borgstrom, A. Singh, B. L. Jordan, G. S. Sukhatme, M. A. Batalin, and W. J. Kaiser. Energy based path planning for a novel cabled robotic system. In *IROS*, 2008.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge UP, March 2004.

I.M. Chao, B.L. Golden, and E.A. Wasil. A fast and effective heuristic for the orienteering problem. *EJOR*, 88:475–489, 1996.

C. Chekuri and M. Pal. A recursive greedy algorithm for walks in directed graphs. In *FOCS*, pages 245–253, 2005.

C. Chekuri, N. Korula, and M. Pal. Improved algorithms for orienteering and related problems. In *SODA*, pages 661–670, 2008.

C. Guestrin, A. Krause, and A.P. Singh. Near-optimal sensor placements in gaussian processes. In *ICML*, 2005.

A. Gupta, R. Krauthgamer, and J.R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *FOCS*, 2003.

G. Hollinger and S. Singh. Proofs and experiments in scalable, near-optimal search by multiple robots. In *RSS*, 2008.

R. A. Houghton. Aboveground forest biomass and the global carbon balance. *Global Change Biology*, 11(6):945–958, 2005.

Andreas Krause and Carlos Guestrin. Nonmyopic active learning of gaussian processes: an exploration-exploitation approach. In *ICML*, 2007.

A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *IPSN*, 2006.

A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. 9:235–284, 2008.

S. Lin. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44:2245–2269, 1965.

A. Meliou, A. Krause, C. Guestrin, and J.M. Hellerstein. Nonmyopic informative path planning in spatio-temporal models. In *AAAI*, pages 602–607, 2007.

G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.

N. Roy, G.J. Gordon, and S. Thrun. Finding approximate pomdp solutions through belief compression. *JAIR*, 23:1–40, 2005.

A. Singh, A. Krause, C. Guestrin, W.J. Kaiser, and M.A. Batalin. Efficient planning of informative paths for multiple robots. In *IJCAI*, pages 2204–2211, 2007.

Sensor planning research challenge: http://www-users.cs.york.ac.uk/g̃rzes/sprc, 2007.

C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *RSS*, 2005.

Sebastian Thrun. Learning occupancy grids with forward sensor models. *Autonomous Robots*, 15:111–127, 2003.

APPENDIX

Proof of Proposition 1. To simultaneously prove submodularity and monotonicity, it suffices to prove that for policies $\pi \subseteq \pi'$ and an *arbitrary* policy x it holds that $U(\pi \cup x) - U(\pi) \geq U(\pi' \cup x) - U(\pi')$. We have, for each $\mathbf{x}_\mathcal{V}$,

$$u((\pi(\mathbf{x}_\mathcal{V}) \cup x(\mathbf{x}_\mathcal{V})), \mathbf{x}_\mathcal{V}) - u(\pi(\mathbf{x}_\mathcal{V}), \mathbf{x}_\mathcal{V}) \geq \\ u((\pi'(\mathbf{x}_\mathcal{V}) \cup x(\mathbf{x}_\mathcal{V})), \mathbf{x}_\mathcal{V}) - u(\pi'(\mathbf{x}_\mathcal{V}), \mathbf{x}_\mathcal{V}),$$

due to monotonicity and submodularity of u. Hence:

$$\int P(\mathbf{x}_\mathcal{V}) \left(u((\pi(\mathbf{x}_\mathcal{V}) \cup x(\mathbf{x}_\mathcal{V})), \mathbf{x}_\mathcal{V}) - u(\pi(\mathbf{x}_\mathcal{V}), \mathbf{x}_\mathcal{V}) \right) d\mathbf{x}_\mathcal{V} \geq \\ \int P(\mathbf{x}_\mathcal{V}) \left(u((\pi'(\mathbf{x}_\mathcal{V}) \cup x(\mathbf{x}_\mathcal{V})), \mathbf{x}_\mathcal{V}) - u(\pi'(\mathbf{x}_\mathcal{V}), \mathbf{x}_\mathcal{V}) \right) d\mathbf{x}_\mathcal{V}. \tag{2}$$

which proves that $U(\pi \cup x) - U(\pi) \geq U(\pi' \cup x) - U(\pi')$.

For proving the (r, γ)-locality property, we first define that two policies pi and π' are at least a distance r apart if, $\pi(\mathbf{x}_\mathcal{V})$ and $\pi'(\mathbf{x}_\mathcal{V})$ are at least a distance r apart for all $\mathbf{x}_\mathcal{V}$. Since u is (r, γ)-local, for two policies π and π', we have for all $\mathbf{x}_\mathcal{V}$

$$u((\pi(\mathbf{x}_\mathcal{V}) \cup \pi'(\mathbf{x}_\mathcal{V})), \mathbf{x}_\mathcal{V}) \geq u(\pi(\mathbf{x}_\mathcal{V}), \mathbf{x}_\mathcal{V}) + \gamma u(\pi'(\mathbf{x}_\mathcal{V}), \mathbf{x}_\mathcal{V}).$$

Therefore:

$$\int P(\mathbf{x}_\mathcal{V}) \left(u((\pi(\mathbf{x}_\mathcal{V}) \cup \pi'(\mathbf{x}_\mathcal{V})), \mathbf{x}_\mathcal{V}) \right) d\mathbf{x}_\mathcal{V} \geq \\ \int P(\mathbf{x}_\mathcal{V}) \left(u(\pi(\mathbf{x}_\mathcal{V}), \mathbf{x}_\mathcal{V}) + \gamma u(\pi'(\mathbf{x}_\mathcal{V}), \mathbf{x}_\mathcal{V}) \right) d\mathbf{x}_\mathcal{V}. \tag{3}$$

i.e. $U(\pi \cup \pi') \geq U(\pi) + \gamma U(\pi')$. Equation (2) and Equation (3) complete the proof. □

Proof of Theorem 3. Let $B' = B/(dim(\mathcal{V}, \mathcal{E})(r+2) + 2)$. Let Π be the set of all sequential policies π of cost $C(\pi) \leq B'$, and $\bar{\mathcal{A}}$ be the collection of sets $\mathcal{A}$ such that the respective nonadaptive policies $\pi_\mathcal{A}$ have cost $C(\pi_\mathcal{A}) \leq B'$. We have that

$$\begin{aligned}
\max_{\pi \in \Pi} \mathbb{E}[U(\pi)] &= \max_{\pi \in \Pi} \int P(\mathbf{x}_\mathcal{V}) u(\pi(\mathbf{x}_\mathcal{V}), \mathbf{x}_\mathcal{V}) d\mathbf{x}_\mathcal{V} \\
&= \max_{\pi \in \Pi} \int P(\mathbf{y}_\mathcal{V}) P(\mathbf{x}_\mathcal{V} \mid \mathbf{y}_\mathcal{V}) u(\pi(\mathbf{x}_\mathcal{V}), \mathbf{x}_\mathcal{V}) d\mathbf{y}_\mathcal{V} d\mathbf{x}_\mathcal{V} \\
&\leq \int P(\mathbf{y}_\mathcal{V}) \max_{\pi \in \Pi} \int P(\mathbf{x}_\mathcal{V} \mid \mathbf{y}_\mathcal{V}) u(\pi(\mathbf{x}_\mathcal{V}), \mathbf{x}_\mathcal{V}) d\mathbf{y}_\mathcal{V} d\mathbf{x}_\mathcal{V} \\
&= \int P(\mathbf{y}_\mathcal{V}) \max_{\mathcal{A} \in \bar{\mathcal{A}}} \int P(\mathbf{x}_\mathcal{V} \mid \mathbf{y}_\mathcal{V}) u(\mathcal{A}, \mathbf{x}_\mathcal{V}) d\mathbf{y}_\mathcal{V} d\mathbf{x}_\mathcal{V} \\
&= \int P(\mathbf{y}_\mathcal{V}) \max_{\mathcal{A} \in \bar{\mathcal{A}}} F(\mathcal{A} \mid \mathbf{y}_\mathcal{V}) \mathbf{x}_\mathcal{V} \\
&= \mathbb{E}_{\mathbf{y}_\mathcal{V}}[\max_{\mathcal{A} \in \bar{\mathcal{A}}} F(\mathcal{A} \mid \mathbf{y}_\mathcal{V})]
\end{aligned}$$

due to Jensen's inequality, the definition of sequential policies and the fact that given all observations $\mathbf{y}_\mathcal{V}$, the optimal policy is to choose a fixed set of locations $\mathcal{A}$ optimized for the observations $\mathbf{y}_\mathcal{V}$. Furthermore,

$$\max_{\pi \in \Pi} U(\pi) \geq \max_{\mathcal{A} \in \bar{\mathcal{A}}} U(\mathcal{A}) = \max_{\mathcal{A} \in \bar{\mathcal{A}}} \mathbb{E}_{\mathbf{y}_\mathcal{V}}[F(\mathcal{A} \mid \mathbf{y}_\mathcal{V})].$$

Hence,

$$\frac{\max_{\pi \in \Pi} U(\pi)}{\max_{\mathcal{A} \in \bar{\mathcal{A}}} U(\pi_\mathcal{A})} \leq \frac{\mathbb{E}_{\mathbf{y}_\mathcal{V}}[\max_{\mathcal{A} \in \bar{\mathcal{A}}} F(\mathcal{A} \mid \mathbf{y}_\mathcal{V})]}{\max_{\mathcal{A} \in \bar{\mathcal{A}}} \mathbb{E}_{\mathbf{y}_\mathcal{V}}[F(\mathcal{A} \mid \mathbf{y}_\mathcal{V})]} = \Gamma.$$

Now, according to Proposition 2, for the solution $\mathcal{A}'$ obtained by pSPIEL$_{OR}$ it holds that

$$F(\mathcal{A}') \geq (1 - e^{-1})\gamma\rho \max_{\mathcal{A}\in\bar{\mathcal{A}}} F(\mathcal{A}).$$

Hence

$$F(\mathcal{A}') \geq \frac{(1 - e^{-1})\gamma\rho}{\Gamma} \max_{\pi\in\Pi} U(\pi).$$

□

Proof of Theorem 4. Let π_i^* be the stage i policy chosen by the optimal solution. Define $\mathcal{O}_i = \cup_{j=1}^{i}\pi_j^*$, with $\mathcal{O}_0 = \emptyset$ and $\mathcal{O}_1 = \pi_1^*$. Similarly, let π_i be the policy chosen by sequential allocation, and set $\mathcal{A}_i = \cup_{j=1}^{i}\pi_j$.

By the assumption of an η-approximation algorithm, the reward collected at stage i can be bounded as:

$$U_{\mathcal{A}_{i-1}}(\pi_i) \geq 1/\eta(U_{\mathcal{A}_{i-1}}(\pi_i^*)).$$

Summing up, after k stages, the total collected reward can be given as:

$$\sum_{i=1}^{k} U_{\mathcal{A}_{i-1}}(\pi_i) \geq 1/\eta(\sum_{i=1}^{k} U_{\mathcal{A}_{i-1}}(\pi_i^*)). \quad (4)$$

Since the left hand side is a telescopic sum, we get:

$$\sum_{i=1}^{k} U_{\mathcal{A}_{i-1}}(\pi_i) = U(\cup_{i=1}^{k}\pi_i) = U(\mathcal{A}_k) \quad (5)$$

On the right hand side (RHS):

$$\begin{aligned} R.H.S. &= 1/\eta(\sum_{i=1}^{k} U_{\mathcal{A}_{i-1}}(\pi_i^*)) \\ &= 1/\eta(\sum_{i=1}^{k}(U(\pi_i^* \cup \mathcal{A}_{i-1}) - U(\mathcal{A}_{i-1}))) \end{aligned}$$

Adding $\mathcal{O}_{i-1}$ to both the terms and using the submodularity property, we get

$$\begin{aligned} R.H.S. &\geq 1/\eta(\sum_{i=1}^{k}(U(\mathcal{O}_i \cup \mathcal{A}_{i-1}) - U(\mathcal{O}_{i-1} \cup \mathcal{A}_{i-1}))) \\ &= 1/\eta\,[U(\mathcal{O}_1) - 0 + U(\mathcal{O}_2 \cup \mathcal{A}_1) - U(\mathcal{O}_1 \cup \mathcal{A}_1) + \\ &\qquad \cdots + U(\mathcal{O}_k \cup \mathcal{A}_{k-1}) - U(\mathcal{O}_{k-1} \cup \mathcal{A}_{k-1})] \end{aligned}$$

Rearranging the terms, we get:

$$R.H.S. \geq 1/\eta \left[U(\mathcal{O}_k \cup \mathcal{A}_{k-1}) - \sum_{i=1}^{k-1}(U(\mathcal{O}_i \cup \mathcal{A}_i) - U(\mathcal{O}_i \cup \mathcal{A}_{i-1}))\right]$$

Using the monotonicity ($U(\mathcal{O}_k \cup \mathcal{A}_{k-1}) \geq U(\mathcal{O}_k)$) and submodularity of U ($U(\mathcal{O}_i \cup \mathcal{A}_i) - U(\mathcal{O}_i \cup \mathcal{A}_{i-1}) \leq U(\mathcal{A}_i) - U(\mathcal{A}_{i-1})$), we get

$$\begin{aligned} R.H.S. &\geq 1/\eta \left[U(\mathcal{O}_k) - \sum_{i=1}^{k-1}(U(\mathcal{A}_i) - U(\mathcal{A}_{i-1}))\right] \\ &= 1/\eta\,[U(\mathcal{O}_k) - U(\mathcal{A}_{k-1})] \end{aligned}$$

Using the monotonicity ($U(\mathcal{A}_k) \geq U(\mathcal{A}_{k-1})$), we get

$$R.H.S. \geq 1/\eta\,[U(\mathcal{O}_k) - U(\mathcal{A}_k)] \quad (6)$$

Substituting Equation (5) and (6) into Equation (4), we get:

$$U(\mathcal{A}_k) \geq 1/\eta\,[U(\mathcal{O}_k) - U(\mathcal{A}_k)],$$

and thus:

$$U(\mathcal{A}_k) \geq 1/(\eta + 1)U(\mathcal{O}_k).$$

resulting in an approximation guarantee of $(1 + \eta)$. □

Proof of Proposition 2. Consider the following lemmas:

Lemma 7 (Proposition 3 in Krause *et al.* [2006]). *Given any path $\mathcal{P}'$ in $\mathcal{G}'$ with weight W, it is possible to find a path $\mathcal{P}$ in $\mathcal{G}$ spanning the same vertices $\mathcal{A}'$, with a total length no more than $\ell(\mathcal{P}')$, and with $F(\mathcal{A}') \geq \gamma W$.*

Lemma 8 (Proposition 4 in Krause *et al.* [2006]). *If the graph $\mathcal{G}$ contains a path $\mathcal{P}*$ of length ℓ^* and value $F(\mathcal{A}^*)$, then there is a path $\mathcal{P}'$ in graph $\mathcal{G}'$ that has length at most*

$$\ell^* \times (dim(\mathcal{V}, \mathcal{E})(r + 2) + 2) \quad (7)$$

and whose expected weight is at least

$$F(\mathcal{A}^*) \times (1 - e^{-1}) \times \rho \quad (8)$$

where ρ is the probability that every node $s \in \mathcal{V}$ is r-padded.

Lemma 8 proves the existence of a path $\mathcal{P}'$ in the graph $\mathcal{G}'$, for which both cost and weight are close to the optimal path $\mathcal{P}^*$ in $\mathcal{G}$. The construction in the proof also guarantees that the path $\mathcal{P}'$ contains at least one cluster center $G_{i,1}$ for some i (or is empty, in which case $\mathcal{P}^*$ is empty). Lemma 7 handles the transfer of the solution to the original graph $\mathcal{G}$. Combining Lemma 7 and Lemma 8, it follows that:

Corollary 9. *Corresponding to the optimal path $\mathcal{P}^*$ with length ℓ^*, there exists a path $\mathcal{P}'$ in graph $\mathcal{G}'$ with cost at most*

$$\ell^*\,(dim(\mathcal{V}, \mathcal{E})(r + 2) + 2) \quad (9)$$

and whose expected weight is at least

$$(1 - e^{-1})\,\gamma\rho F(\mathcal{A}^*). \quad (10)$$

Now, consider the following lemma for the approximation algorithm for path planning:

Lemma 10 (Theorem 1.1 by Chekuri *et al.* [2008]). *For any fixed $\delta > 0$, there is an algorithm with running time $n^{O(1/\delta^2)}$ which gives a (2 + δ) approximation for orienteering in undirected graphs.*

Combining Lemma 10 on the modular approximation graph $\mathcal{G}'$, with Corollary 9, there exists a polynomial time approximation algorithm that outputs a path starting at node s_1 and finishing at node s_B with approximation guarantee of $(2 + \epsilon)(1 - e^{-1})\gamma\rho$. Since the *tour-opt* heuristics applied for path smoothening will only decrease the path length of the selected path while keeping the nodes in the selected path, Proposition 2 follows.

□

Proof of Proposition 5. Note that

$$\Gamma \leq \mathbb{E}_{\mathbf{y}_\mathcal{V}} \left[\max_{\mathcal{A} \in \bar{\mathcal{A}}} \frac{F(\mathcal{A} \mid \mathbf{y}_\mathcal{V})}{F(\mathcal{A})} \right].$$

For any set of visited locations $\mathcal{A}$ let $\mathcal{W}_i(\mathcal{A}) = \bigcup_{s \in \mathcal{A}} \mathcal{W}_{i,s}$. Hence, $F(\mathcal{A} \mid obs) = \sum_{s' \in \mathcal{W}(\mathcal{A})} P(\mathcal{X}_s' \mid obs)$. Hence

$$\frac{F(\mathcal{A} \mid \mathbf{y}_\mathcal{V})}{F(\mathcal{A})} \leq \max_{s' \in \mathcal{W}_i(\mathcal{A})} \frac{P(\mathcal{X}_{s'} \mid \mathbf{y}_\mathcal{V})}{P(\mathcal{X}_{s'})}.$$

□

Proof of Proposition 6. The proof for the exploitation contribution u_2 to the utility is analogous to the proof of Proposition 5. For the exploration contribution, there is no adaptivity gap, as shown by Krause and Guestrin [2007]. □