Informative Path Planning with Limited Adaptivity

Rayen Tan
University of Michigan

Rohan Ghuge
Georgia Institute of Technology

Viswanath Nagarajan
University of Michigan

Abstract

We consider the *informative path planning* (IPP) problem in which a robot interacts with an *uncertain* environment and gathers information by visiting locations. The goal is to minimize its expected travel cost to cover a given submodular function. Adaptive solutions, where the robot incorporates all available information to select the next location to visit, achieve the best objective. However, such a solution is resource-intensive as it entails recomputing after every visited location. A more practical approach is to design solutions with a small number of adaptive "rounds", where the robot recomputes only once at the start of each round. In this paper, we design an algorithm for IPP parameterized by the number k of adaptive rounds, and prove a smooth tradeoff between k and the solution quality (relative to fully adaptive solutions). We validate our theoretical results by experiments on a real road network, where we observe that a few rounds of adaptivity suffice to obtain solutions of cost almost as good as fully-adaptive ones.

1 INTRODUCTION

We consider the *informative path planning* (IPP) problem in which a robot interacts with an *uncertain* environment and gathers information by visiting locations. The informative path planning problem has been widely studied, and has applications in information gathering (Singh, Krause, Guestrin, et al. 2006), object detection (Platt et al. 2011), and manipulating a robot arm for tasks like pushing a button or grasping (Javdani et al. 2014). We discuss two applications of IPP.

Proceedings of the 27$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

First, consider the following disaster management application. Suppose that an autonomous unmanned aerial vehicle (UAV) is searching for a lost victim (Lim, Hsu, and Lee 2016). The UAV acquires new information on the victim's location by using onboard sensors, and the goal is to plan a search strategy in order to find the victim as fast as possible. Another application of IPP, arising in information gathering, is in monitoring algae biomass in a lake (Dhariwal et al. 2006). It is not economical to cover the lake sufficiently with *static* sensors, and instead one wants to plan a route for a robotic boat (carrying a sensor) to move to various locations in the lake to gather information.

We note that both these applications involve *stochastic submodular optimization*: the uncertainty stems from not knowing the underlying state of the world (the victim's true location, or the concentration of pollutants in the lake), and progress (eliminating possible locations from consideration, or collecting information from different parts of the lake) can be captured using a submodular function (see Lim, Hsu, and Lee (2016) and Singh, Krause, Guestrin, et al. (2006) for details). In most works, stochastic submodular optimization is restricted over a set domain; that is, the goal is to select some subset to optimize the expected objective. However, we are interested in settings where the robot interacts with the environment by visiting different sensing location; in other words, the robot's decisions are constrained to form a *path* (rather than an arbitrary subset). So, the goal in IPP is to *minimize its expected travel cost to cover a given submodular function* (see § 1.1 for a formal definition).

Solutions to such a stochastic problem are sequential decision processes, making them highly adaptive: at each step, the robot incorporates all available information to select the next location to visit. This process continues until the submodular function is covered. However, such a solution entails recomputing after every visited location, which can be resource-intensive. So, fully adaptive solutions may not be feasible in practical situations. This motivates the design of solutions with a small number of adaptive "rounds", where the robot recomputes only once at the start of each round. Such solutions strike a balance between achieving the best objective and keeping resource utilization low. For example, from an energy efficiency perspective: a robot often operates with limited battery capacity, and the need for frequent re-computations after each visited location can lead to substantially faster energy consumption. Additionally, the process of incorporating observed data and preparing it for re-computation is not instantaneous as it involves data gathering/analysis, and may lead to increased computational time. By keeping the rounds of adaptivity small, we aim to mitigate such operational challenges. We note that the tradeoff between rounds of adaptivity and solution quality is not new, and has been studied in various streams of literature (see § 2). We make the following contributions.

1. We design an algorithm for IPP parameterized by the number k of adaptive rounds, and prove a smooth tradeoff between k and the solution quality (relative to fully adaptive solutions).

2. We consider separately an important special case of IPP: path planning for hypothesis identification (Lim, Hsu, and Lee 2016), and obtain a better performance guarantee via a more efficient algorithm.

3. Finally, we run computational experiments on a real road network dataset and previously-used instances of hypothesis identification. For these instances, we observe that with 2 rounds of adaptivity, the cost is on average within 50% and 12% of the fully adaptive cost, respectively. Moreover, the 2-round algorithm is on average 15 times faster than the fully adaptive one.

1.1 Definitions

An instance of IPP is given by the tuple $(X, r, d, M, \mathcal{D}, O, f)$, where X corresponds to a finite set of n sensing locations, r is the initial location of the robot, and d is a metric on $X \cup \{r\}$. We assume throughout that d is symmetric and satisfies triangle inequality. Here $M = \{1, \ldots, m\}$ denotes a finite set of *hypotheses* or *scenarios*. We take the Bayesian approach, and use p_ω to denote the prior probability of each hypothesis $\omega \in M$, where $\sum_{\omega \in M} p_\omega = 1$. The set O denotes all possible observations: each location $v \in X$ realizes to a random observation in O. The distribution $\mathcal{D}$ specifies the probability p_ω of each hypothesis $\omega \in M$ as well as the observations $\{\omega(v) \in O : v \in X\}$ at all locations *under hypothesis* ω. The true hypothesis ω^* is drawn from M according to the distribution $\mathcal{D}$. While the prior distribution $\mathcal{D}$ is known to the algorithm, the true hypothesis ω^* is initially unknown. When location $v \in X$ is visited, the robot observes $\omega^*(v) \in O$ and can use this information to update its priors. In this paper, we assume that our observations are *noiseless*, i.e. for a fixed hypothesis w^*, the observation $w^*(v)$ is deterministic.

If the robot has visited a subset $S \subseteq X$ of locations and observed $o_v \in O$ at each $v \in S$ then the set $\{(v, o_v) : v \in S\} \subseteq X \times O$ is called a *partial realization*. We use $\psi \subseteq X \times O$ to denote a generic partial realization; if we want to additionally specify the locations S contained in ψ then we use the notation $\psi(S)$. We define $\psi_\omega(X) := \{(v, \omega(v)) : v \in X\}$ to denote the partial realization associated with hypothesis ω at all locations. We say that hypothesis ω is *compatible* with a partial realization ψ, denoted $\omega \sim \psi$, if $\psi \subseteq \psi_\omega(X)$.

Let $\Psi = 2^{X \times O}$ be the power-set of all location-realization pairs. Note that Ψ contains every partial realization. Not every subset in Ψ corresponds to an actual partial realization, but using the full power-set Ψ makes the definition of our utility function cleaner. In particular, function $f : \Psi \to \mathbb{Z}_+$ is a monotone submodular set function. Formally, we say that f is *submodular* if whenever $A \subseteq B \subseteq \Psi$ and $e \notin B$, we have $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$, and we say that f is *monotone* if whenever $A \subseteq B$, we have $f(A) \leq f(B)$. Let $Q \in \mathbb{Z}_+$ be the maximal value of the function. We say that a partial realization $\psi \in \Psi$ *covers* function f if $f(\psi) = Q$. We assume that $f(\psi_\omega(X)) = Q$ for each hypothesis $\omega \in M$. In other words, if we visit all locations then the function f will be covered, irrespective of the true hypothesis ω^*.

Let $\Pi = (r, v_1, \ldots, v_\ell, r)$ be any tour that starts and ends at r; we abuse notation and also use Π to denote the set of locations visited. All tours in this paper will begin/end at r: we will not state this condition each time (to avoid clutter). We say that tour Π *covers* hypothesis ω if, and only if, $f(\psi_\omega(\Pi)) = Q$. The goal in IPP is to design a tour Π (possibly adaptively) that covers the true hypothesis ω^*. An equivalent condition is that the observed partial realization $\psi(\Pi)$ at the end of the tour must satisfy $f(\psi(\Pi)) = Q$. The objective is to minimize the expected distance $\mathbb{E}[d(\Pi)]$ of the tour Π, where the expectation is taken over ω^*. We note that an adaptive tour Π decides on the next location to visit based on the observations at all previous locations. We are interested in solutions that have limited adaptivity as defined next.

Definition 1.1. *For an integer $k \geq 1$, a* **k-round** *solution proceeds in k rounds of adaptivity. In each round $\ell \in \{1, \ldots, k\}$, the solution specifies a tour on all remaining locations and visits them in this order until some stopping rule (at which point it starts the next round). The tour in round ℓ can depend on the* *observations seen in rounds* $1, \ldots, \ell - 1$.

In a k-round solution, tour re-computation only occurs at the start each round, which happens at most k times. Setting $k = 1$ in Definition 1.1 gives us a *non-adaptive* tour that does not have to recompute after it starts. On the other hand, setting $k \to \infty$ (effectively $k = n$) gives us a *fully adaptive* tour, that recomputes after each visited location. Having more rounds potentially leads to a smaller objective value, so fully adaptive solutions have the least objective value. Our performance guarantees are relative to an optimal fully adaptive solution; let OPT denote this solution and its cost. The *k-round-adaptivity gap* is defined as follows:

$$\sup_{\text{instance } I} \frac{\mathbb{E}[\text{cost of best } k\text{-round solution on } I]}{\mathbb{E}[\text{cost of best fully adaptive solution on } I]}.$$

In formulating IPP, we required the solution to be a *tour* originating from r. We note that one could also ask for a *path* originating from r, which is allowed to end at any location as long as the function f is covered. All our results also apply to this path variant, formalized in the following proposition.

Proposition 1.1. *Any α-approximation algorithm for the tour version of* IPP *gives a 2α-approximation to the path version of* IPP.

Indeed, by symmetry and triangle inequality, the cost of an optimal tour is at most 2 times the cost of an optimal path, and any α-approximate solution for the tour version of IPP is also feasible for the path version.

A noteworthy special case of IPP is informative path planning for hypothesis identification ($\texttt{IPP}-\texttt{H}$). $\texttt{IPP}-\texttt{H}$ has function f counting the number of *eliminated* hypotheses: this is monotone and submodular. Covering the function corresponds to eliminating all but one hypothesis, which is the same as identifying the true hypothesis. A more efficient algorithm for $\texttt{IPP}-\texttt{H}$ is explored in § 4.

1.2 Example: UAV Search

We formalize the UAV search application from Lim, Hsu, and Lee (2016) as an instance of $\texttt{IPP}-\texttt{H}$ to demonstrate the use of our model. This problem models a UAV searching for a lost victim in an area modeled as an $N \times N$ grid. The UAV can be at "high" or "low" altitudes. At high altitudes, the UAV can use a long range sensor to determine whether the missing person is in the 3×3 grid around its location. At low altitudes, the UAV can use a more accurate sensor to determine whether the current location contains the victim.

An instance of the UAV search problem is given by the tuple $(X, r, d, M, \mathcal{D}, O)$. The set X corresponds to the $2 \times N \times N$ sensing locations (a sensing location is a cell in the $N \times N$ grid in either high or low altitudes) and r is the initial location of the UAV. The distance metric $d(i, j)$ is the Manhattan distance between sensing locations i and j. Scenarios M correspond to possible locations of the victim on the $N \times N$ grid, and p_ω is the prior probability that the victim is at $\omega \in M$. For each $v \in X$, there is a set $S_v \subseteq X$ of locations that can be "sensed" from v. We set $O = \{0, 1\}$ where an observation of 1 at location v implies that the victim is in one of the locations S_v. Finally, note that the function f in $\texttt{IPP}-\texttt{H}$ counts the number of scenarios that are eliminated.

2 RELATED WORK

It is known that metrics of informativeness in several domains (for example, sensor placement (Krause and Guestrin 2005) and target search (Hollinger, Singh, et al. 2009)) exhibit *submodularity*. Submodular set function optimization has been studied extensively (Wolsey 1982; Nemhauser, Wolsey, and Fisher 1978), and has also been extended to optimizing over paths (Chekuri and Pál 2005; Călinescu and Zelikovsky 2005). Consequently, any approximation algorithm for *submodular path orienteering* (Chekuri and Pál 2005) can be used to plan a path for a robot in order to maximize a submodular function of the visited locations. Singh, Krause, Guestrin, et al. (2006) provided an approach for extending any single robot algorithm to the multi-robot setting, with a (nearly) matching approximation guarantee.

Submodular optimization (over sets) has been extended to the stochastic setting in a number of works, e.g., Asadpour and Nazerzadeh (2016), Golovin and Krause (2017), Im, Nagarajan, and Zwaan (2016), Grammel et al. (2016), and Gupta, Nagarajan, and Singla (2017). Recent works (Agarwal, Assadi, and Khanna 2019; Esfandiari, Karbasi, and Mirrokni 2021; Ghuge, Gupta, and Nagarajan 2021) are particularly relevant to us: these papers establish trade-offs between *rounds of adaptivity* and the approximation factor for stochastic submodular cover problems, where one wants to select a *subset* to cover a submodular function (there are different settings with independent, scenario-based and adaptive-submodularity conditions). Our work extends the results from Ghuge, Gupta, and Nagarajan (2021) for scenario-based distributions to the case of optimizing over *paths* in a metric.

Stochastic submodular optimization over paths has also received significant attention. IPP has been studied in robotics and related fields, and many heuristic approaches have been proposed to solve the problem. For example, in Hollinger, Englot, et al. (2013), a minimum-cost tour is constructed on "informative" sensing locations, and in Hollinger, Mitra, and Sukhatme (2011), the idea is to search for a strategy over a finite planning horizon. An adaptive approach appeared in Singh, Krause, and Kaiser (2009): their algorithm re-plans every step using a non-adaptive information path planning algorithm.

The special case of $\mathtt{IPP-H}$ has itself been studied widely. This appears in Gupta, Nagarajan, and Ravi (2017) as the "isolation problem" enroute to obtaining approximation algorithms for the *adaptive traveling salesman* problem. Gupta, Nagarajan, and Ravi (2017) obtained a fully-adaptive $O(\log^2 n \log m)$-approximation algorithm for $\mathtt{IPP-H}$. Lim, Hsu, and Lee (2015) and Lim, Hsu, and Lee (2016) obtained similar algorithms for IPP; these hold for a slightly more general definition involving adaptive-submodularity (Golovin and Krause 2017). When applied to IPP, the algorithms in Lim, Hsu, and Lee (2015) and Lim, Hsu, and Lee (2016) yield a fully adaptive $O\left(\log^{2+\epsilon}(n) \cdot \log(m) \cdot \log\left(1/p_{\min}\right)\right)$-approximation algorithm; here $p_{min} \leq 1/m$ is the minimum probability of any hypothesis. Navidi, Kambadur, and Nagarajan (2020) obtained an improved $O\left(\log^{2+\epsilon}(n) \cdot \log(m)\right)$-approximate fully adaptive algorithm for IPP.

The tradeoff between rounds of adaptivity and solution quality has also been considered in other contexts. Gao et al. (2019), Esfandiari, Karbasi, Mehrabian, et al. (2021), and Agarwal, Ghuge, and Nagarajan (2022) study online learning problems, where observations are made in batches. Balkanski, Breuer, and Singer (2018), Balkanski and Singer (2018), Balkanski, Rubinstein, and Singer (2019), and Chekuri and Quanrud (2019) study deterministic submodular optimization, where function queries are batched. However, the techniques used in these papers are completely different from ours.

The role of adaptivity was first formally studied in Dean, Goemans, and Vondrák (2008) for the stochastic knapsack problem. Since then, it has been has been extensively studied for various stochastic optimization problems such as stochastic submodular maximization (Asadpour and Nazerzadeh 2016; Gupta, Nagarajan, and Singla 2017; Bradac, Singla, and Zuzic 2019), stochastic matching (Bansal, Gupta, et al. 2012; Behnezhad, Derakhshan, and Hajiaghayi 2020), set cover (Goemans and Vondrák 2006), submodular cover (Agarwal, Assadi, and Khanna 2019; Ghuge, Gupta, and Nagarajan 2021), k-TSP (Jiang et al. 2020), intersections of matroids (Gupta and Nagarajan 2013) and orienteering (Guha and Munagala 2009; Gupta, Krishnaswamy, et al. 2015; Bansal and Nagarajan 2015).

3 k-ROUND ALGORITHM FOR IPP

In this section, we prove the following result.

Theorem 3.1. *For any integer $k \geq 1$ and $\epsilon > 0$, there is a k-round adaptive algorithm for* IPP *with cost at most $\mathcal{O}\left(\log^{2+\epsilon}(n) \cdot m^{1/k} \cdot (\log m + k \log Q)\right)$ times the cost of an optimal adaptive algorithm.*

A key component of our algorithm is a non-adaptive algorithm to solve a *partial cover* version of IPP. Formally, an instance of the partial cover version of IPP is the same as an instance of IPP with an additional parameter $\delta \in (0, 1]$. Now, the goal is to visit a set of locations T that realize to $\psi(T) \in \Psi$ such that either (i) number of compatible scenarios $|\{\omega \in M : \psi(T) \subseteq \psi_\omega\}| < \delta m$, or (ii) the function f is fully covered, i.e., $f(\psi(T)) = Q$. The k-round algorithm for IPP will then recursively solve the partial cover version with carefully chosen values for the parameter δ.

An important subroutine in our algorithm is the following *deterministic* problem.

Definition 3.1 (Ratio Submodular Orienteering (RSO))**.** *Given a metric d on locations $X \cup \{r\}$ and a monotone submodular function $g : 2^X \to \mathbb{Z}_+$, find an r-tour T that maximizes the ratio $\frac{g(T)}{d(T)}$, where $g(T)$ is the function value on the nodes of T and $d(T)$ is the total distance in T.*

This problem is NP-hard, but there are polylogarithmic approximation ratios known:

Theorem 3.2 (Călinescu and Zelikovsky 2005)**.** *For any constant $\epsilon > 0$ there is an $O(\log^{2+\epsilon} n)$-approximation algorithm for* RSO *in runtime $n^{O(1/\epsilon)}$.*

If one allows for quasi-polynomial time $n^{O(\log n)}$ then a better $O(\log n)$-approximation algorithm is known (Chekuri and Pál 2005). It is also hard to approximate to a factor better than $O(\log^{1-\epsilon} n)$ (Halperin and Krauthgamer 2003.)

We first prove the following.

Theorem 3.3. *There is a non-adaptive algorithm for the partial cover version of* IPP *with expected cost $\mathcal{O}\left(\frac{\rho}{\delta} \log\left(\frac{Q}{\delta}\right)\right)$ times the cost of the optimal adaptive solution for* IPP*, where ρ is the best approximation guarantee for ratio submodular orienteering.*

The algorithm creates a *pre-planned* (non-adaptive) tour; that is, without knowing the realizations at the locations. We find that iteratively selecting tours that maximize a carefully-defined score function (see Equation (1)) works well; however, selecting such tours turns out to be an NP-hard problem. So, at each step we pick a tour that *approximately* maximizes the score function, which is subsequently appended to the non-adaptive tour: this process continues until all locations are included in the non-adaptive tour, or we can conclude that the number of compatible scenarios after visiting the already selected locations will be less than δm (see Definition 3.2). We note that the score of a tour (roughly) measures the progress we can makes towards (i) eliminating scenarios and (ii) covering function f on visiting the tour. Crucially, we prove that the numerator of this score function corresponds to a monotone submodular function (see Lemma 3.4). So, we can use an approximation algorithm for RSO to optimize the score. Before we state the score function, we need some definitions.

Definition 3.2. *For any $S \subseteq X$, let $\mathcal{H}(S)$ denote the partition $\{Y_1, \cdots, Y_\ell\}$ of the scenarios M where all scenarios in a part have the same realization for the locations in S. Let $\mathcal{Z} := \{Y \in \mathcal{H}(S) : |Y| \geq \delta|M|\}$ be the set of "large" parts.*

Consider scenarios ω_1 and ω_2. According to Definition 3.2, ω_1 and ω_2 belong to the same part of $\mathcal{H}(S)$ if and only if $\omega_1(v) = \omega_2(v)$ for all $v \in S$; i.e., visiting the locations in S leads to the *same* partial realization under either scenario ω_1 or ω_2. After observing the realization of S, the set of compatible scenarios must be one of the parts in $\mathcal{H}(S)$. Also note that $|\mathcal{Z}| \leq \frac{1}{\delta}$ as each part in $\mathcal{Z}$ has at least $\delta|M|$ scenarios.

Definition 3.3. *For any location $v \in X$ and subset $Z \subseteq M$ of scenarios, consider the partition of Z based on the realization of v. Let $B_v(Z) \subseteq Z$ be the largest cardinality part, and define $L_v(Z) := Z \setminus B_v(Z)$.*

The above definition is used to quantify the "information gain" of visiting a single location. If the realized scenario $\omega^* \in L_v(Z)$, then we can eliminate at least half the scenarios in Z by visiting location v. For any part $Z \in \mathcal{H}(S)$, note that the partial realizations $\psi_\omega(S)$ are identical for all $\omega \in Z$: we use $\psi_Z(S) \subseteq X \times O$ to denote this partial realization.

Let Π denote the non-adaptive tour constructed so far in our algorithm, and let S be the set of locations in Π. The score (1) of a new tour T is computed by considering two notions of progress for each $Z \in \mathcal{Z}$:

- *Information gain* $\sum_{\omega \in L_T(Z)} p_\omega$, measures the total probability of the scenarios that belong to $L_v(Z)$ for some $v \in T$.
- *Relative function gain* $\sum_{\omega \in Z} p_\omega \cdot \frac{f(\psi_Z(S) \cup \psi_\omega(T)) - f(\psi_Z(S))}{Q - f(\psi_Z(S))}$ measures the expected relative gain obtained by visiting locations in tour T (expectation is w.r.t. scenarios in Z).

The overall score of tour T is the sum of these terms (over all parts in $\mathcal{Z}$) normalized by the distance $d(T)$ of the tour. Note that the score of a tour is computed only using "large" parts $\mathcal{Z}$. This is because if the realization of S corresponds to any other part then the number of compatible scenarios would be less than δm (and the partial cover algorithm would have terminated). We show that the numerator of (1) is a monotone and submodular function (see Lemma 2.2 in the full version (Tan, Ghuge, and Nagarajan 2023)).

Algorithm 1 Partial Covering Algorithm PCA$((X, r, d, M, \mathcal{D}, O, f), \delta)$

$S \leftarrow \emptyset$, $\Pi \leftarrow \emptyset$
while $S \neq X$ **do**
  Define $\mathcal{H}(S)$, $\mathcal{Z}$ and $L_v(Z)$ as in Definitions 3.2 and 3.3
  if $\mathcal{Z}$ is empty **then** **break**
  Select tour T that ρ-approximately maximizes:

$$\text{score}(T) = \frac{1}{d(T)} \cdot \sum_{Z \in \mathcal{Z}} \left(\sum_{\omega \in L_T(Z)} p_\omega + \sum_{\omega \in Z} p_\omega \cdot \frac{f(\psi_Z(S) \cup \psi_\omega(T)) - f(\psi_Z(S))}{Q - f(\psi_Z(S))} \right) \quad (1)$$

  where $L_T(Z) = \cup_{v \in T} L_v(Z)$.
  $S \leftarrow S \cup T$, $\Pi \leftarrow \Pi \circ T$
$R \leftarrow \emptyset$, $\psi(R) \leftarrow \emptyset$, $H \leftarrow M$.
while $|H| \geq \delta m$ and $f(\psi) < Q$ **do**
  $T \leftarrow$ first tour in Π not yet visited
  $\psi(T) \leftarrow$ realization of vertices in T
  $R \leftarrow R \cup T$, $\psi(R) \leftarrow \psi(R) \cup \psi(T)$
  $H \leftarrow \{\omega \in H : \omega \sim \psi(R)\}$; that is, set of compatible scenarios
return visited locations R, partial realization $\psi(R)$ and compatible scenarios H.

Lemma 3.4. *Let*

$$g(T) = \sum_{Z \in \mathcal{Z}} \left(\sum_{\omega \in L_T(Z)} p_\omega + \sum_{\omega \in Z} p_\omega \cdot \frac{f(\psi_Z(S) \cup \psi_\omega(T)) - f(\psi_Z(S))}{Q - f(\psi_Z(S))} \right),$$

where $S \subseteq X$ is some fixed subset. Then g is monotone and submodular.

Note that $\text{score}(T) = g(T)/d(T)$, and since $g(T)$ is monotone and submodular, we can use an approximation algorithm for RSO to optimize the score. Once the non-adaptive tour Π, which itself is a concatenation of many smaller tours, is specified, the algorithm starts by visiting tours in this order until (i) the number of compatible scenarios drops below δm, or (ii) the realized function value equals Q. Note that in case (ii),

Algorithm 2 k-round adaptive algorithm for IPP, k-ADAP$(\mathcal{I} = (X, r, d, M, \mathcal{D}, O, f), k)$

1: Run PCA$(\mathcal{I}, m^{-1/k})$ for the first round. Let R denote the set of locations visited. Let ψ and H denote the partial realization and set of compatible scenarios respectively.
2: Define the residual submodular function $f_\psi(\phi) = f(\psi \cup \phi) - f(\psi)$, and define the distribution D_H by conditioning on the remaining scenarios H.
3: Solve k-ADAP$(\widehat{\mathcal{I}} = (X \setminus R, r, d, H, \mathcal{D}_H, O, f_\psi), k-1)$

the function is fully covered. See Algorithm 1 for a formal description of the non-adaptive algorithm.

We recursively use this non-adaptive partial cover algorithm to get a k-round solution for IPP. The first round involves setting $\delta = m^{-1/k}$ in PCA. At the end of round #1, let R be the set of locations visited, $\psi = \psi(R)$ be the partial realization observed, and $H \subseteq M$ be the compatible scenarios. Then, we can condition on the scenarios in H, and define a "residual" function $f_\psi : \Psi \to \mathbb{Z}_+$ as $f_\psi(\phi) = f(\psi \cup \phi) - f(\psi)$, which is also monotone and submodular. Finally, we recurse on this residual function f_ψ to get a $k-1$ round solution. See Algorithm 2 for a formal description. We formalize this discussion in the following result.

Theorem 3.5. *Algorithm 2 is a k-round algorithm for* IPP *with expected cost $\mathcal{O}\left(\rho \cdot m^{1/k} \cdot (\log m + k \log Q)\right)$ times the optimal fully adaptive cost. Here, m is the number of scenarios and ρ is the approximation guarantee for* RSO.

Combined with the approximation algorithm for submodular orienteering (Theorem 3.2), this proves Theorem 3.1.

The proof of Theorem 3.5 can be found in Tan, Ghuge, and Nagarajan (2023, § 2). The rest of this section is devoted to the proof of Theorem 3.3.

3.1 Proof of Theorem 3.3

For the analysis, we denote our non-adaptive policy, and its (random) cost as NA. Similarly, we use OPT to refer to an optimal fully adaptive policy and its (random) cost. We refer to the cumulative cost incurred by either policy as *elapsed time*. We define constants β (specified later) and $L := \log\left(\frac{Q}{\delta}\right)$. Next, we define terms that are used to track the progress of OPT and NA respectively.

- $o(t) := \mathbf{P}\,(\texttt{OPT}$ does not terminate by time $t)$
- $a(t) := \mathbf{P}\,(\texttt{NA}$ does not terminate by time $\beta L t)$

Observe that $o(t)$ and $a(t)$ are non-increasing functions of t, and $o(0) = a(0) = 1$. We can view $o(t)$ and $a(t)$ as "non-completion" probabilities of OPT and NA respectively. The following key lemma relates these non-completion probabilities.

Lemma 3.6. *For any $i \geq 0$, we have*

$$\frac{\beta\delta}{4\rho} \cdot \sum_{j \geq i} \left(\frac{a(j+1) - 2 \cdot o(j+1)}{j+1} \right) \leq a(i). \quad (2)$$

Using this lemma, we can prove Theorem 3.3.

Proof of Theorem 3.3. Using the integral identity for expectations, we can write the expected cost of our non-adaptive policy as follows.

$$\mathbb{E}[\texttt{NA}] = \int_0^\infty \mathbf{P}(\texttt{NA} > t)dt = \int_0^\infty a\left(\frac{t}{\beta L}\right) dt$$
$$= \beta L \int_0^\infty a(t)dt$$

where the final equality follows by applying a change of variables. Since $a(t)$ is non-increasing in t, we have

$$\mathbb{E}[\texttt{NA}] = \beta L \int_0^\infty a(t)dt \leq \beta L \sum_{i \geq 0} a(i) = \beta L \cdot A \quad (3)$$

where we set $A = \sum_{i \geq 0} a(i)$. Similarly, we let $O = \sum_{i \geq 0} o(i)$, and sum (2) over $i \geq 0$ to obtain

$$\frac{4\rho}{\beta\delta} \cdot A = \frac{4\rho}{\beta\delta} \cdot \sum_{i \geq 0} a(i) \geq \sum_{i \geq 0} \sum_{j \geq i} \left(\frac{a(j+1) - 2 \cdot o(j+1)}{j+1} \right)$$
$$= \sum_{j \geq 1} (a(j) - 2 \cdot o(j)) \geq A - 2O$$

where the final inequality uses $a(0) = o(0) = 1$. On rearranging the above inequality, we obtain

$$A \leq \frac{2\beta\delta}{\beta\delta - 4\rho} \cdot O. \quad (4)$$

Finally, we can write the expected cost of OPT in terms of O as follows.

$$O - 1 = \sum_{j \geq 1} o(j) \leq \int_0^\infty o(t)dt = \mathbb{E}[\texttt{OPT}]. \quad (5)$$

where the inequality holds since $o(\cdot)$ is non-increasing. On combining Equations (3),(4), and (5) we get

$$\mathbb{E}[\texttt{NA}] \leq \beta L \cdot A \leq \frac{2\beta^2 L\delta}{\beta\delta - 4\rho} \cdot O \leq \frac{2\beta^2 L\delta}{\beta\delta - 4\rho} \cdot (\mathbb{E}[\texttt{OPT}] + 1)\,.$$

Setting $\beta = \frac{8\rho}{\delta}$ implies that $\mathbb{E}[\texttt{NA}] \leq \frac{32L\rho}{\delta} \cdot (\mathbb{E}[\texttt{OPT}] + 1)$. We note that the $+1$ term can be eliminated by a straightforward scaling argument: note that, for any $b \geq 1$, if all costs are scaled by b (resulting in OPT and NA being scaled by b), we get $\mathbb{E}[\texttt{NA}] \leq \frac{32L\rho}{\delta} \cdot \left(\mathbb{E}[\texttt{OPT}] + \frac{1}{b}\right)$, thus implying that for large enough b, we get $\mathbb{E}[\texttt{NA}] \leq \frac{32L\rho}{\delta} \cdot \mathbb{E}[\texttt{OPT}]$, as desired. ☐

The proof of Lemma 3.6 is found in Tan, Ghuge, and Nagarajan (2023, § 2). We note that our k-round adaptivity gap is nearly best possible. As shown in Ghuge, Gupta, and Nagarajan (2021), there are instances of IPP (even on star metrics) where every k-round solution has cost at least $\Omega(\frac{m^{1/k}}{k\log m})$ times the optimal adaptive cost. So, we need at least $\frac{\log m}{\log\log m}$ rounds to achieve any poly-logarithmic approximation.

4 IMPROVED ALGORITHM FOR HYPOTHESIS IDENTIFICATION

Here, we consider an important special case of IPP: path planning for hypothesis identification ($\mathtt{IPP}-\mathtt{H}$). An instance is given by the tuple $(X, r, d, M, \mathcal{D}, O)$. Here, X is the set of sensing locations, r is the root location and d is a metric on $X \cup \{r\}$. Set $M = \{1, \ldots, m\}$ is a finite set of *hypotheses/scenarios* with probabilities $\{p_\omega\}_{\omega\in M}$. The set O denotes the possible observations; each $v \in X$ realizes to a random observation in O. The distribution $\mathcal{D}$ specifies the probability p_ω of each scenario $\omega \in M$ as well as the observations $\{\omega(v) \in O : v \in X\}$ at all locations under scenario ω. The true hypothesis/scenario $\omega^* \in M$ according to the distribution $\mathcal{D}$; however ω^* is initially unknown to the algorithm. The goal is to identify ω^* by visiting locations at the minimum expected distance. When location $v \in X$ is visited, the robot observes $\omega^*(v) \in O$ and can use this information to update its priors.

We obtain the following improved result for $\mathtt{IPP}-\mathtt{H}$.

Theorem 4.1. *For any integer $k \geq 1$, there is a randomized k-round adaptive algorithm for hypothesis identification with cost $\mathcal{O}\left(\log^2(n)\cdot m^{1/k}\cdot k\cdot\log m\right)$ times the optimal adaptive cost.*

In order to cast $\mathtt{IPP}-\mathtt{H}$ as a special case of IPP, we define a submodular function $f : 2^{X\times O} \to \mathbb{R}_+$ as follows. For each $v \in X$ and $o \in O$, let $E_{v,o} \subseteq M$ be the set of hypotheses that are *incompatible* with observation o at location v. Note that if we observe o at location v then we must have $\omega^* \notin E_{v,o}$. Now, we define, $\forall S \subseteq X$ and $o_v \in O$ for $v \in S$:

$$f(\{(v, o_v) : v \in S\}) = \Big|\bigcup_{v\in S} E_{v,o_v}\Big|. \tag{6}$$

Note that this is exactly the number of incompatible hypotheses after having visited locations S and observed o_v at each $v \in S$. It is easy to see that f is monotone and submodular: it is a set coverage function. Clearly, ω^* is identified precisely when this number is $m - 1$. So, we set our target $Q = m - 1$.

Recall that at each step of the partial covering algorithm PCA, we need to solve an instance of RSO with the following objective function:

$$g(T) = \sum_{Z\in\mathcal{Z}}\Bigg(\sum_{\omega\in L_T(Z)} p_\omega + \sum_{\omega\in Z} p_\omega\cdot\frac{f(\psi_Z(S)\cup\psi_\omega(T)) - f(\psi_Z(S))}{Q - f(\psi_Z(S))}\Bigg),$$

$\forall T \subseteq X$. Above, $S \subseteq X$ is a fixed subset. This is exactly the criterion in (1).

In the special case of $\mathtt{IPP}-\mathtt{H}$, we show below that all RSO instances correspond to the simpler *ratio group Steiner* problem:

Definition 4.1 (Ratio Group Steiner)**.** *An instance consists of a metric (V, d) with nodes V and distances $d : V\times V \to \mathbb{R}_+$. There is a special root node $r \in V$ and k* groups*, where each group $i \in [k]$ is associated with a subset $S_i \subseteq V$ and weight $w_i \geq 1$. We want to find a tour τ originating from r that minimizes the ratio of its distance to the weight of groups covered.*

Moreover, we provide a simpler algorithm for this problem that is faster in practice (see Tan, Ghuge, and Nagarajan (2023)[§ 3.1]).

Theorem 4.2. *There is a randomized $O(\log^2 n)$-approximation algorithm for ratio group Steiner, where $n = |V|$ is the number of nodes.*

We now construct an instance of ratio group Steiner corresponding to the RSO instance when function f is given by (6). The metric and root remains the same. The groups and weights are as follows:

- *Groups for information gain (1st term in g).* For each $Z \in \mathcal{Z}$ and scenario $\omega \in Z$ there is a group consisting of nodes $\{v \in X : \omega \in L_v(X)\}$ with weight p_ω.
- *Groups for function gain (2nd term in g).* For each $Z \in \mathcal{Z}$, note that the compatible scenarios after observing partial realization $\psi_Z(S)$ is exactly Z. So, $f(\psi_Z(S)) = m - |Z|$ and $\forall R \subseteq X \times O$,

  $$f(\psi_Z(S)\cup R) - f(\psi_Z(S)) = \left|Z \cap \left(\cup_{(v,o)\in R} E_{v,o}\right)\right|.$$

  Hence, for any $\omega \in Z$ and $\forall T \subseteq X$, we have

  $$\frac{f(\psi_Z(S)\cup\psi_\omega(T)) - f(\psi_Z(S))}{Q - f(\psi_Z(S))} = \frac{1}{|Z|-1}\sum_{\theta\in Z}\mathbf{1}\left[\theta \in \cup_{v\in T} E_{v,\psi_\omega(v)}\right].$$

  Moreover, note that $\theta \in \cup_{v\in T} E_{v,\psi_\omega(v)}$ iff the outcomes at v under scenarios ω and θ are different, i.e., $\psi_\omega(v) \neq \psi_\theta(v)$. Now, we introduce a group for each $Z \in \mathcal{Z}$ and scenarios $\omega, \theta \in Z$ with nodes $\{v \in X : \psi_\omega(v) \neq \psi_\theta(v)\}$ and weight $\frac{p_\omega}{|Z|-1}$.

So, it follows that for any $T \subseteq X$, the total weight of covered groups is $g(T)$. Hence, this RSO instance reduces to the ratio group Steiner problem.

Completing the proof of Theorem 4.1. Using the above ratio group Steiner instance and Theorem 4.2, we obtain a $\rho = O(\log^2 n)$ approximation algorithm for RSO instances arising from $\mathtt{IPP}-\mathtt{H}$. Combined with Theorem 3.5, this implies Theorem 4.1.

Tighter Approximation Using More Rounds. Using our partial covering algorithm (Theorem 3.3) and a different measure of progress in each round (as in Theorem 6.7 of Ghuge, Gupta, and Nagarajan (2021)), we can also obtain $2k$-round algorithms with better approximation guarantees of $\mathcal{O}\left(\log^{2+\epsilon}(n) \cdot m^{1/k} \cdot \log(Qm)\right)$ for IPP and $\mathcal{O}\left(\log^2(n) \cdot m^{1/k} \cdot \log m\right)$ for $\mathtt{IPP}-\mathtt{H}$ (see Tan, Ghuge, and Nagarajan (2023)[§ A] for details). Setting the number of rounds to $O(\log m)$, we then get approximation guarantees of $O\left(\log^{2+\epsilon}(n) \cdot \log(Qm)\right)$ and $O\left(\log^2(n) \cdot \log m\right)$ for IPP and $\mathtt{IPP}-\mathtt{H}$ respectively. These approximation ratios match the previous-best approximation ratios for these problems, even for fully-adaptive algorithms (Navidi, Kambadur, and Nagarajan 2020; Gupta, Nagarajan, and Ravi 2017). In fact, $\mathtt{IPP}-\mathtt{H}$ and IPP generalize the *group Steiner tree* problem (Garg, Konjevod, and Ravi 2000), for which the best known approximation ratio is $O\left(\log^2(n) \cdot \log m\right)$; there is also an $\Omega(\log^{2-\epsilon} n)$ hardness of approximation (Halperin and Krauthgamer 2003).

5 COMPUTATIONAL RESULTS

We provide a summary of computational results of our k-round algorithm for path planning for hypothesis identification ($\mathtt{IPP}-\mathtt{H}$). We test our algorithm on two sets of instances: UAV search, (that were also used in Lim, Hsu, and Lee (2016)), and a real-world road network (Li et al. 2005). We consider the path version of $\mathtt{IPP}-\mathtt{H}$, where the robot/UAV does not have to return to the root r at the end; by Proposition 1.1, all our results apply to this path version. We also skip returning to the root at the end of each round: the solution goes directly from the last node of a round to the first node of the next round. (By triangle inequality, the modified solution is at least as good as the solution from Algorithm 2.) We note that the experiments in Lim, Hsu, and Lee (2016) were also for the path version. These experiments show that our algorithm, using a small number of rounds, performs well when compared with fully-adaptive algorithms, and also demonstrates the computational benefits of limited adaptivity.

Instances. We use a real-world dataset for the California road network, which describes connections between "points of interests" in California, and contains $21,047$ nodes and $21,691$ edges. We observe that a majority of the nodes in the network are degree-2 nodes; so we can combine such nodes and only keep nodes with degree 3 or more. The resulting network has $1,365$ nodes and $1,991$ edges (see Figure 1).

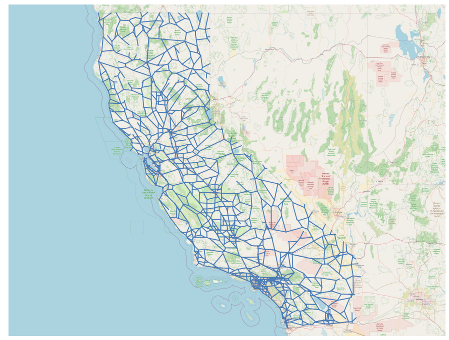


Figure 1: California road network.

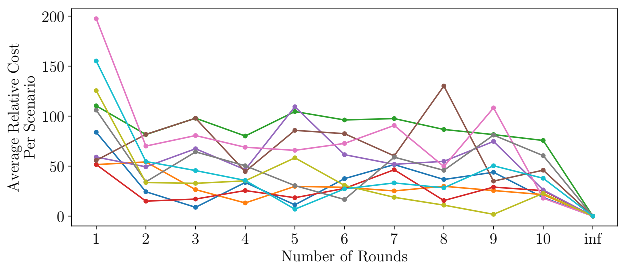
We use this network to generate 10 instances of $\mathtt{IPP}-\mathtt{H}$ as follows. We set $m = 50$ and generate the scenarios using the independent cascade model (ICM) (see Tan, Ghuge, and Nagarajan (2023, § 4.1) for details). Let scenario i correspond to set S_i: given scenario i, the robot receives a feedback of 1 from node v if $v \in S_i$, and 0 otherwise. The robot begins near the node with the highest centrality measure, and the goal is to identify the underlying scenario with minimum expected cost. The ICM is parameterized by edge probabilities that determines the likelihood of influence: we use $p \in \{0.6, 0.62, 0.64, 0.66, 0.68\}$, and generate two instances for each p.

We also use some synthetic instances, motivated by UAV search, which is based on prior work (Lim, Hsu, and Lee 2016). These instances are smaller and display more geometric (grid like) structure. Previously, Lim, Hsu, and Lee (2016) ran their adaptive $\mathtt{IPP}-\mathtt{H}$ algorithms on an 8×8 grid instance. In addition to this instance, we also evaluate our algorithms on larger 9×9 and 10×10 grid instances. More details of these UAV search instances are in Tan, Ghuge, and Nagarajan (2023, § 4).

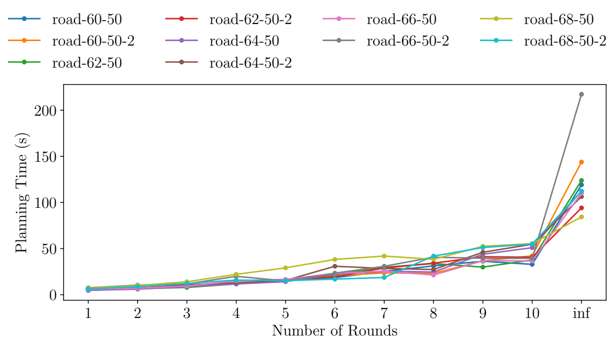
Results. We run our k-round algorithm for $k \in \{1, \ldots, 10\} \cup \{\mathtt{inf}\}$, where inf denotes the fully adaptive algorithm (practically, setting $k = n$ effectively gives a fully adaptive algorithm). For each instance, we record the cost to identify each scenario. We normalize the cost achieved by the k-round algorithm against the adaptive algorithm (per scenario), yielding an average relative cost (ARC), given by the following equation:

$$\text{ARC} = \mathbb{E}_i \left[\frac{k\text{-round}(i) - \texttt{ADAP}(i)}{\texttt{ADAP}(i)} \right] \times 100\%$$

where k-round(i) (resp. $\texttt{ADAP}(i)$) denotes the cost incurred by the k-round (resp. fully adaptive) algorithm for scenario i. We also report the average CPU time (over scenarios) taken by the k-round algorithm. Since the tour in the first round is the same across all scenarios, we amortize its computation time across scenarios.



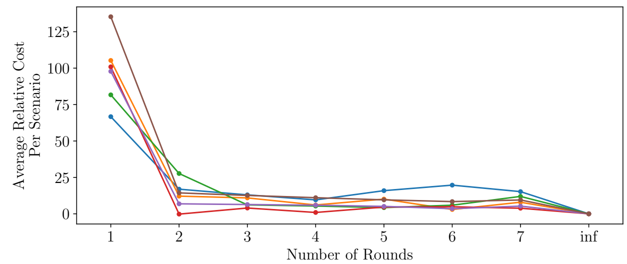
(a) ARC v/s # rounds



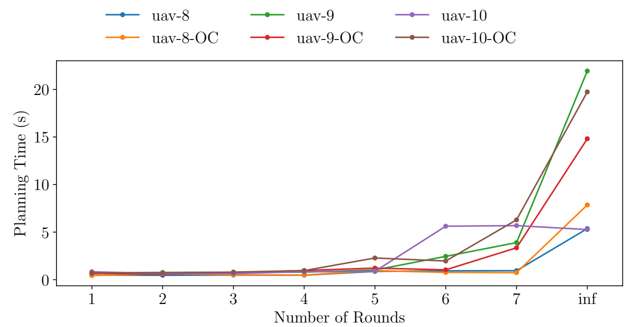
(b) Computation time v/s # rounds

Figure 2: Results for road network instances

We report results for the road network and UAV search instances in Figures 2 and 3 respectively. We observe that, with an increase in the number of rounds, the cost (ARC) typically decreases and computation time increases. However, the trend of ARC sometimes shows an increased cost with more rounds: this can be attributed to the randomness in our algorithm (Theorem 4.1), which is due to use of probabilistic tree embedding. One notable result is for the UAV instance $8 - \texttt{OC}$, which was the only instance tested in Lim, Hsu, and Lee (2016): even with 2 rounds, our solution cost is about 30% less than the fully-adaptive solution found in Lim, Hsu, and Lee (2016). We observe that using 2 rounds of adaptivity, the cost is on average within 50% (for road network instances) and within 12% (for UAV instances) of the fully adaptive cost. Notably, the 2-round algorithm is 15 times faster than the fully adaptive one (averaging over instances and scenarios). In light of these computational results, using 2-3 rounds of adaptivity gives a good trade-off between cost and runtime.



(a) ARC v/s # rounds



(b) Computation time v/s # rounds

Figure 3: Results for UAV search instances

6 CONCLUSION

In this paper, we design an algorithm for the informative path planning problem parameterized by the number k of adaptive rounds, and prove a smooth trade-off between k and the solution quality. Our computational experiments corroborate our theory, showing that a few rounds of adaptivity suffice to get solutions comparable to fully adaptive ones, while providing a significant benefit in computational time. We leave open the question of designing algorithms under uncertainty of measurements: can we design algorithms with limited adaptivity when measurements may be imprecise or even incorrect with some probability?

Acknowledgement

This research was supported in part by NSF grants CMMI-1940766 and CCF-2006778.

References

Agarwal, Arpit, Sepehr Assadi, and Sanjeev Khanna (2019). "Stochastic Submodular Cover with Limited Adaptivity". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2019)*, pp. 323–342.

Agarwal, Arpit, Rohan Ghuge, and Viswanath Nagarajan (2022). "Batched Dueling Bandits". In: *International Conference on Machine Learning (ICML)*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 89–110.

Asadpour, Arash and Hamid Nazerzadeh (2016). "Maximizing Stochastic Monotone Submodular Functions". In: *Manag. Sci.* 62.8, pp. 2374–2391.

Balkanski, Eric, Adam Breuer, and Yaron Singer (2018). "Non-monotone Submodular Maximization in Exponentially Fewer Iterations". In: *Advances in Neural Information Processing Systems*, pp. 2359–2370.

Balkanski, Eric, Aviad Rubinstein, and Yaron Singer (2019). "An Exponential Speedup in Parallel Running Time for Submodular Maximization without Loss in Approximation". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 283–302.

Balkanski, Eric and Yaron Singer (2018). "Approximation Guarantees for Adaptive Sampling". In: *Proceedings of the 35th International Conference on Machine Learning*, pp. 393–402.

Bansal, N., A. Gupta, et al. (2012). "When LP Is the Cure for Your Matching Woes: Improved Bounds for Stochastic Matchings". In: *Algorithmica* 63.4, pp. 733–762.

Bansal, Nikhil and Viswanath Nagarajan (2015). "On the adaptivity gap of stochastic orienteering". In: *Math. Program.* 154.1-2, pp. 145–172.

Behnezhad, Soheil, Mahsa Derakhshan, and MohammadTaghi Hajiaghayi (2020). "Stochastic matching with few queries: (1-ϵ) approximation". In: *Procceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1111–1124.

Bradac, Domagoj, Sahil Singla, and Goran Zuzic (2019). "(Near) Optimal Adaptivity Gaps for Stochastic Multi-Value Probing". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Vol. 145, 49:1–49:21.

Călinescu, Gruia and Alexander Zelikovsky (2005). "The Polymatroid Steiner Problems". In: *J. Comb. Optim.* 9.3, pp. 281–294.

Chekuri, Chandra and Martin Pál (2005). "A recursive greedy algorithm for walks in directed graphs". In: *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pp. 245–253.

Chekuri, Chandra and Kent Quanrud (2019). "Parallelizing greedy for submodular set function maximization in matroids and beyond". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pp. 78–89.

Dean, B. C., M. X. Goemans, and J. Vondrák (2008). "Approximating the Stochastic Knapsack Problem: The Benefit of Adaptivity". In: *Math. Oper. Res.* 33.4, pp. 945–964.

Dhariwal, Amit et al. (2006). "Networked Aquatic Microbial Observing System". In: *IEEE ICRA*.

Esfandiari, Hossein, Amin Karbasi, Abbas Mehrabian, et al. (2021). "Regret Bounds for Batched Bandits". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.8, pp. 7340–7348.

Esfandiari, Hossein, Amin Karbasi, and Vahab Mirrokni (2021). "Adaptivity in Adaptive Submodularity". In: *Proceedings of 34th Conference on Learning Theory*. Vol. 134. PMLR, pp. 1823–1846.

Gao, Zijun et al. (2019). "Batched Multi-armed Bandits Problem". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc.

Garg, Naveen, Goran Konjevod, and R. Ravi (2000). "A Polylogarithmic Approximation Algorithm for the Group Steiner Tree Problem". In: *J. Algorithms* 37.1, pp. 66–84.

Ghuge, Rohan, Anupam Gupta, and Viswanath Nagarajan (2021). "The Power of Adaptivity for Stochastic Submodular Cover". In: *International Conference on Machine Learning*. PMLR, pp. 3702–3712.

Goemans, Michel and Jan Vondrák (2006). "Stochastic Covering and Adaptivity". In: *LATIN 2006: Theoretical Informatics*. Springer Berlin Heidelberg, pp. 532–543.

Golovin, Daniel and Andreas Krause (2017). "Adaptive Submodularity: A New Approach to Active Learning and Stochastic Optimization". In: *CoRR* abs/1003.3967.

Grammel, Nathaniel et al. (2016). "Scenario submodular cover". In: *International Workshop on Approximation and Online Algorithms*. Springer, pp. 116–128.

Guha, Sudipto and Kamesh Munagala (2009). "Multi-armed Bandits with Metric Switching Costs". In: *Automata, Languages and Programming, 36th Internatilonal Colloquium (ICALP)*, pp. 496–507.

Gupta, Anupam, Ravishankar Krishnaswamy, et al. (2015). "Running Errands in Time: Approximation Algorithms for Stochastic Orienteering". In: *Math. Oper. Res.* 40.1, pp. 56–79.

Gupta, Anupam and Viswanath Nagarajan (2013). "A Stochastic Probing Problem with Applications". In: *IPCO*, pp. 205–216.

Gupta, Anupam, Viswanath Nagarajan, and R. Ravi (2017). "Approximation Algorithms for Optimal Decision Trees and Adaptive TSP Problems". In: *Math. Oper. Res.* 42.3, pp. 876–896.

Gupta, Anupam, Viswanath Nagarajan, and Sahil Singla (2017). "Adaptivity Gaps for Stochastic

Probing: Submodular and XOS Functions". In: *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1688–1702.

Halperin, Eran and Robert Krauthgamer (2003). "Polylogarithmic inapproximability". In: *ACM Symposium on Theory of Computing*. Ed. by Lawrence L. Larmore and Michel X. Goemans. ACM, pp. 585–594.

Hollinger, Geoffrey A., Brendan Englot, et al. (2013). "Active planning for underwater inspection and the benefit of adaptivity". In: *The International Journal of Robotics Research* 32, pp. 3–18.

Hollinger, Geoffrey A., Urbashi Mitra, and Gaurav S. Sukhatme (2011). "Active Classification: Theory and Application to Underwater Inspection". In: *International Symposium of Robotics Research*.

Hollinger, Geoffrey A., Sanjiv Singh, et al. (2009). "Efficient Multi-robot Search for a Moving Target". In: *The International Journal of Robotics Research* 28, pp. 201–219.

Im, Sungjin, Viswanath Nagarajan, and Ruben van der Zwaan (2016). "Minimum Latency Submodular Cover". In: *ACM Trans. Algorithms* 13.1, 13:1–13:28.

Javdani, Shervin et al. (2014). "Near Optimal Bayesian Active Learning for Decision Making". In: *International Conference on Artificial Intelligence and Statistics*.

Jiang, Haotian et al. (2020). "Algorithms and Adaptivity Gaps for Stochastic k-TSP". In: *11th Innovations in Theoretical Computer Science Conference (ITCS)*. Vol. 151, 45:1–45:25.

Krause, Andreas and Carlos Guestrin (2005). "Near-Optimal Nonmyopic Value of Information in Graphical Models". In: *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pp. 324–331.

Li, Feifei et al. (Aug. 2005). "On Trip Planning Queries in Spatial Databases". In: vol. 3633, pp. 273–290.

Lim, Zhan Wei, David Hsu, and Wee Sun Lee (2015). "Adaptive Stochastic Optimization: From Sets to Paths". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc.

– (2016). "Adaptive informative path planning in metric spaces". In: *The International Journal of Robotics Research* 35.5, pp. 585–598.

Navidi, Fatemeh, Prabhanjan Kambadur, and Viswanath Nagarajan (2020). "Adaptive Submodular Ranking and Routing". In: *Oper. Res.* 68.3, pp. 856–877.

Nemhauser, George L., Laurence A. Wolsey, and Marshall L. Fisher (1978). "An analysis of approximations for maximizing submodular set functions—I". In: *Mathematical Programming* 14, pp. 265–294.

Platt, Robert et al. (2011). "Simultaneous Localization and Grasping as a Belief Space Control Problem". In: *15th international symposium on Robotics Research*.

Singh, Amarjeet, Andreas Krause, Carlos Guestrin, et al. (2006). "Efficient Planning of Informative Paths for Multiple Robots". In: *International Joint Conference on Artificial Intelligence*.

Singh, Amarjeet, Andreas Krause, and William J. Kaiser (2009). "Nonmyopic Adaptive Informative Path Planning for Multiple Robots". In: *International Joint Conference on Artificial Intelligence*.

Tan, Rayen, Rohan Ghuge, and Viswanath Nagarajan (2023). "Informative Path Planning with Limited Adaptivity". In: *CoRR* abs/2311.12698.

Wolsey, L.A. (1982). "An analysis of the greedy algorithm for the submodular set covering problem". In: *Combinatorica* 2.4, pp. 385–393.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]

   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]