# Adaptive Multi-Robot Wide-Area Exploration and Mapping

Kian Hsiang Low[†], John M. Dolan[†§], and Pradeep Khosla[†§]
Department of Electrical and Computer Engineering[†], Robotics Institute[§]
Carnegie Mellon University
5000 Forbes Avenue Pittsburgh PA 15213 USA
{bryanlow, jmd}@cs.cmu.edu, pkk@ece.cmu.edu

## ABSTRACT

The exploration problem is a central issue in mobile robotics. A complete terrain coverage is not practical if the environment is large with only a few small hotspots. This paper presents an adaptive multi-robot exploration strategy that is novel in performing both wide-area coverage and hotspot sampling using non-myopic path planning. As a result, the environmental phenomena can be accurately mapped. It is based on a dynamic programming formulation, which we call the *Multi-robot Adaptive Sampling Problem* (MASP). A key feature of MASP is in covering the entire adaptivity spectrum, thus allowing strategies of varying adaptivity to be formed and theoretically analyzed in their performance; a more adaptive strategy improves mapping accuracy. We apply MASP to sampling the Gaussian and log-Gaussian processes, and analyze if the resulting strategies are adaptive and maximize wide-area coverage and hotspot sampling. Solving MASP is non-trivial as it comprises continuous state components. So, it is reformulated for convex analysis, which allows discrete-state monotone-bounding approximation to be developed. We provide a theoretical guarantee on the policy quality of the approximate MASP (aMASP) for using in MASP. Although aMASP can be solved exactly, its state size grows exponentially with the number of stages. To alleviate this computational difficulty, anytime algorithms are proposed based on aMASP, one of which can guarantee its policy quality for MASP in real time.

## Categories and Subject Descriptors

G.1.6 [**Optimization**]: convex programming; G.3 [**Probability and Statistics**]: stochastic processes; I.2.8 [**Problem Solving, Control Methods, and Search**]: dynamic programming; I.2.9 [**Robotics**]: autonomous vehicles

## General Terms

Algorithms, Performance, Design, Experimentation, Theory

## Keywords

multi-robot exploration and mapping, adaptive sampling, active learning, Gaussian process, log-Gaussian process, non-myopic path planning

**Cite as:** Adaptive Multi-Robot Wide-Area Exploration and Mapping, Kian Hsiang Low, John M. Dolan, and Pradeep Khosla, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 23-30.


## 1. INTRODUCTION

The problem of exploring an unknown environment is a central issue in mobile robotics. Typically, it requires sampling the entire terrain. However, a complete terrain coverage is not practical in terms of resource costs if the environment is large with only a few small, dynamic "hotspots", and the robot sensing range is limited. Such an environment arises in two important real-world applications: (a) planetary exploration such as geologic reconnaissance and mineral prospecting [9], and (b) environment and ecological monitoring such as monitoring of ocean phenomena (e.g., algal bloom) [7], forest ecosystems [13], pollution, and contamination. In these applications, it is often necessary to sample the hotspots for detailed analysis and extraction. At the same time, the environment has to be adequately covered to locate these hotspots as well as map the phenomena accurately. To reduce human effort and risk, it is desirable to build robot teams that can perform these tasks.

An important issue in designing such a robot team is the *exploration strategy*: how do the robots decide where to explore next? This paper presents an adaptive model-based exploration strategy that is novel in performing both wide-area coverage and hotspot sampling, and covering the entire adaptivity spectrum. In contrast, all other model-based strategies are non-adaptive and achieve only wide-area coverage. Our strategy can also plan non-myopic multi-robot paths, which are more desirable than greedy or single-robot paths. These characteristics distinguish our approach from the existing robot exploration strategies and are discussed in greater detail with the related work below:

**Wide-Area Coverage vs. Hotspot Sampling** Exploration strategies [7, 12, 13, 16, 17, 18] that emphasize wide-area coverage aim to improve the mapping accuracy of the environmental phenomena. On the other hand, strategies that focus on locating and sampling hotspots [9] may not be tailored to satisfy this objective. In contrast, our proposed approach can tackle both tasks simultaneously.

**Model- vs. Design-Based Strategies** In design-based strategies [9, 13, 18], the selection of sampling locations for exploration is constrained by the sampling design, which is not devised to consider resource costs. As a result, the locations have to be chosen by the strategy first before minimizing the resource costs to sample them. This entails "sunk" costs in motion; the exploration paths have to traverse terrain that do not require sampling to reach the selected locations. Furthermore, some strategies [13, 18] require mul-

tiple "passes" through the region of interest such that new locations are adaptively selected and sampled in each pass. Modifying the strategy to involve resource costs may invalidate the estimators associated with the strategy.

We instead use a model-based strategy [7, 12, 16, 17], which assumes a certain environmental model and selects sampling locations to reduce its uncertainty. Resource cost minimization or constraints may be applied to the selection process and the resulting strategy is optimal in these constraints. In contrast to the strategies [7, 12, 16] that use a parametric model, our approach utilizes a non-parametric model, which does not require any assumptions on the distribution underlying the sampling data. In particular, we model the environmental phenomena as Gaussian [17] and log-Gaussian processes.

**Adaptive vs. Non-Adaptive Sampling Strategies** Adaptive sampling refers to strategies [9, 13, 18] in which the procedure for selecting locations to be included in robot paths depends on the sampling data observed during exploration. On the other hand, non-adaptive sampling strategies [7, 12, 16, 17] have no such dependence. When the environmental phenomena are smoothly varying, non-adaptive strategies are known to perform well [18]. However, if the environment contains hotspots, adaptive sampling can exploit the clustering phenomena to map the environmental field more accurately than non-adaptive sampling. In contrast to the above schemes, the adaptivity of our proposed strategy can be varied (Section 2.3); by increasing adaptivity, its mapping accuracy can be improved.

**Greedy vs. Non-Myopic Path Planning Strategies** In contrast to greedy strategies [12, 17] that face the local minima problem, our strategy generates non-myopic robot paths [16]. Non-myopic paths usually approximate the optimal trajectories better, but incur higher computational cost.

**Single- vs. Multi-Robot Strategies** In contrast to single-robot exploration strategies [12, 13, 16], our strategy has to coordinate the exploration of multiple robots like those in [7, 9, 17, 18]. A robot team can potentially complete the task faster than a single robot and is also robust to failures by providing redundancy.

# 2. MULTI-ROBOT ADAPTIVE SAMPLING PROBLEM (MASP)

## 2.1 Terminology and Notation

Let $\mathcal{X}$ be the domain of the environmental phenomena corresponding to a finite, discretized set of grid cell locations (i.e., cell centers). Let $Z_x$ be a random variable representing the unobserved quantity at an arbitrary location $x \in \mathcal{X}$; its realized observed/sampled quantity will be denoted by $z_x$. Let the data of $m$ sampled locations $d_0$ be an ordered list of pairs $\langle x_i, z_{x_i} \rangle$ for $i = 1, \ldots, m$. So, the addition of the data of $n$ new sampled locations to $d_0$ results in $d_n = [\langle x_1, z_{x_1} \rangle, \ldots, \langle x_{m+n}, z_{x_{m+n}} \rangle]$. Let $\mathbf{x}_{d_n}$ and $\mathbf{z}_{d_n}$ denote vectors comprising the $x$ and $z_x$ components of the data $d_n$ respectively (i.e., $\mathbf{x}_{d_n} = (x_1, \ldots, x_{m+n})$ and $\mathbf{z}_{d_n} = (z_{x_1}, \ldots, z_{x_{m+n}})$). A robot path $\mathcal{P}$ corresponds to a sequence of $l$ locations. When a robot travels between two locations $x$ and $y$, it incurs a motion cost of $C(x, y)$. The cost $C(\mathcal{P})$ of a robot path $\mathcal{P} = \langle x_1, \ldots, x_l \rangle$ is defined as the sum of the motion costs along the path, i.e., $\sum_{i=2}^{l} C(x_{i-1}, x_i)$.

We will define more clearly the notion of adaptivity in an exploration strategy, which is crucial to understanding MASP. Suppose that the data $d_0$ has been sampled previously and $n$ new locations are to be selected for sampling. Formally, an exploration strategy is *strictly adaptive* if its procedure to select *each* new sampling location $x_{m+i+1}$ for $i = 0, \ldots, n-1$ depends only on the previously sampled data $d_i$. A strategy is *non-adaptive* [7, 12, 16, 17] if its procedure to select each new sampling location $x_{m+i+1}$ for $i = 0, \ldots, n-1$ is independent of $z_{x_{m+1}}, \ldots, z_{x_{m+n}}$. Hence, all the $n$ new locations can be selected prior to exploration without needing to observe any new sampling data. By hybridizing the two, a *partially adaptive* [9, 13, 18] strategy results, that is, its procedure to select *each batch* of $j$ new locations $x_{m+ij+1}, \ldots, x_{m+ij+j}$ for $i = 0, \ldots, n/j - 1$ [1] depends only on the previously sampled data $d_{ij}$. When $j = 1$ $(n)$, this strategy becomes strictly adaptive (non-adaptive). So, by increasing the number of new locations in each batch, the adaptivity of the hybrid strategy decreases.

## 2.2 Problem Formulation

The exploration objective is to collect the "best" sampling data from the planned robot paths that maximizes the mapping accuracy of the environmental field. To achieve this, we use the mean-squared error criterion as a measure of the spatial mapping uncertainty. Given the previously sampled data $d_0$, a predictor $\hat{Z}(x_0, d_0)$ of the unobserved quantity $Z_{x_0}$ at location $x_0$ achieves the mean-squared error $\mathbb{E}\{[Z_{x_0} - \hat{Z}(x_0, d_0)]^2 | d_0\}$. Then, the uncertainty in mapping the environmental field of domain $\mathcal{X}$ with predictor $\hat{Z}(x_0, d_0)$ can be represented by the sum of mean-squared errors over all locations in $\mathcal{X}$ (i.e., $\sum_{x_0 \in \mathcal{X}} \mathbb{E}\{[Z_{x_0} - \hat{Z}(x_0, d_0)]^2 | d_0\}$). Using the best unbiased predictor $\hat{Z}(x_0, d_0) \stackrel{\text{def}}{=} \mathbb{E}[Z_{x_0} | d_0]$ (i.e., it achieves the lowest mean-squared error among all unbiased predictors), the mean-squared error at each location $x_0$ can be reduced to the conditional variance $\sigma^2_{Z_{x_0}|d_0} \stackrel{\text{def}}{=} \text{var}[Z_{x_0} | d_0]$. This results in the spatial mapping uncertainty $\sum_{x_0 \in \mathcal{X}} \sigma^2_{Z_{x_0}|d_0}$, which will be used in formulating the exploration problems below.

In essence, the multi-robot exploration problem involves selecting new sampling locations for the robot exploration paths that provide the least amount of spatial mapping uncertainty. One typical way of doing this would be by choosing all the new locations to be added to the existing sample (say, $d_0$) that minimize the expectation of the sum of posterior variances over all locations in $\mathcal{X}$:

$$\min_{\mathcal{P}_1, \ldots, \mathcal{P}_k} \mathbb{E}\{ \sum_{x_0 \in \mathcal{X}} \sigma^2_{Z_{x_0}|d_0, D_{\mathcal{P}_1}, \ldots, D_{\mathcal{P}_k}} \mid d_0 \} \qquad (1)$$

subject to the motion constraint $C(\mathcal{P}_i) \leq B$ for robot $i = 1, \ldots, k$ where $D_{\mathcal{P}_i}$ is an ordered list of pairs $\langle x, Z_x \rangle$ constructed from the path $\mathcal{P}_i$. To interpret (1), the selection of new locations $\mathbf{x}_{D_{\mathcal{P}_1}, \ldots, D_{\mathcal{P}_k}}$ to be included in the robot paths $\mathcal{P}_1, \ldots, \mathcal{P}_k$ does not depend on the previously sampled data along the paths. Hence, this problem formulation is non-adaptive.

On the other hand, the new locations can also be selected sequentially such that the selection of subsequent locations to be included in the robot paths depends on the previously sampled data along the paths. This form of exploration

[1] to simplify exposition, we assume that $n$ is divisible by $j$.

can be achieved by modeling MASP using *Dynamic Programming* (DP) with the following value functions, which represent the amount of spatial mapping uncertainty:

$$\begin{aligned} V_i(d_{ki}) &= \min_{\mathbf{a}_i \in \mathcal{A}(\mathbf{x}_{d_i^k})} \mathbb{E}[V_{i+1}(d_{ki}, D_{i+1}^k) \mid d_{ki}] \\ V_n(d_{kn}) &= \sum_{x_0 \in \mathcal{X}} \sigma^2_{Z_{x_0}|d_{kn}} \end{aligned} \tag{2}$$

for $i = 0, \ldots, n-1$ where the realized data $d_i^k$ is an ordered list of the last $k$ pairs of $d_{ki}$, $\mathbf{x}_{d_i^k} = (x_{m+k(i-1)+1}, \ldots, x_{m+ki})$ is a vector of $k$ robot locations denoting the current state of the robot team, $\mathcal{A}(\mathbf{x}_{d_i^k})$ is the action space of the robot team (i.e., a finite set of joint actions) given its current state $\mathbf{x}_{d_i^k}$, the unobserved data $D_{i+1}^k$ is an ordered list of pairs $\langle x_\ell, Z_{x_\ell} \rangle$ for $\ell = m + ki + 1, \ldots, m + ki + k$, and $\mathbf{x}_{D_{i+1}^k}$ is the next state produced by the deterministic transition function of the robot team $T(\mathbf{a}_i, \mathbf{x}_{d_i^k})$ based on its current action $\mathbf{a}_i$ and state $\mathbf{x}_{d_i^k}$. To interpret (2), the selection of subsequent locations $\mathbf{x}_{D_{i+1}^k}$ to be included in the robot paths $\mathcal{P}_1, \ldots, \mathcal{P}_k$ depends on the previously sampled data $d_{ki}$ along the paths, which makes this problem formulation adaptive. Hence, the random data $D_1^k, \ldots, D_n^k$ corresponds to the sample that is to be realized by the robot paths $\mathcal{P}_1, \ldots, \mathcal{P}_k$. The above motion constraints on the robots also apply in this problem.

Let $\boldsymbol{\pi} = \langle \boldsymbol{\pi}_0(d_0), \ldots, \boldsymbol{\pi}_{n-1}(d_{k(n-1)}) \rangle$ denote the action policy of the robot team such that $\boldsymbol{\pi}_i(d_{ki}) \stackrel{\text{def}}{=} \mathbf{a}_i \in \mathcal{A}(\mathbf{x}_{d_i^k})$. By solving (2), the optimal value $V_0(d_0)$ can be obtained together with the corresponding optimal action policy $\boldsymbol{\pi}^*$ where

$$\boldsymbol{\pi}_i^*(d_{ki}) = \arg\min_{\mathbf{a}_i \in \mathbf{A}(\mathbf{x}_{d_i^k})} \mathbb{E}[V_{i+1}(d_{ki}, D_{i+1}^k) \mid d_{ki}] \ . \tag{3}$$

From (3), the optimal action $\boldsymbol{\pi}_0^*(d_0)$ can be determined prior to exploration since data $d_0$ is known. However, each action rule $\boldsymbol{\pi}_i^*(d_{ki}), i = 1, \ldots, n-1$ defines the actions to take in response to the data $d_{ki}$, part of which (i.e., $[\langle x_{m+1}, z_{x_{m+1}} \rangle, \ldots, \langle x_{m+ki}, z_{x_{m+ki}} \rangle]$) will only be observed during exploration. Given the starting locations (say, $\mathbf{x}_{d_0^k}$) of the robots, their paths $\mathcal{P}_1, \ldots, \mathcal{P}_k$ can also be derived by applying the transition function $T(.,.)$ to the optimal action policy $\boldsymbol{\pi}^*$.

In some exploration tasks, a tradeoff may ensue between adaptivity and task execution cost: for example, the primary environmental variable to be mapped is sometimes associated with highly correlated auxiliary variables, which may be cheaper to sample at a higher spatial resolution or more reliable for planning the exploration paths. So, if an auxiliary variable is used in MASP for path planning instead, an adaptive strategy has to incur the extra cost of sampling the auxiliary variable during exploration, which is not required by non-adaptive exploration.

## 2.3 Advantage of Adaptive Exploration

Increasing adaptivity can improve mapping accuracy (i.e., lower spatial mapping uncertainty) as shown below:

THEOREM 2.1. *Define the value functions of $j$-MASP as*

$$\begin{aligned} V_i^j(d_{kij}) &= \min_{\mathbf{a}_{ij}, \ldots, \mathbf{a}_{ij+j-1}} \mathbb{E}[V_{i+1}^j(d_{kij}, D_{ij+1}^k, \ldots, D_{ij+j}^k) \mid d_{kij}] \\ V_{n/j}^j(d_{kn}) &= \sum_{x_0 \in \mathcal{X}} \sigma^2_{Z_{x_0}|d_{kn}} \end{aligned} \tag{4}$$

*for $i = 0, \ldots, n/j - 1$ where $j$ is the number of robot team actions per stage, $\mathbf{x}_{D_{ij+l}^k} := T(\mathbf{a}_{ij+l-1}, \mathbf{x}_{D_{ij+l-1}^k})$ for $l = 1, \ldots, j$ such that $D_{ij}^k := d_{ij}^k$, and assume $n$ is divisible by $j$. Then, $V_0^j(d_0)$ is monotonically increasing in $j$.*

To elaborate, each stage of $j$-MASP in (4) consists of a minimum followed by an expectation. As the number $j$ of robot team actions per stage decreases, the number of locations sampled per stage decreases, which implies an increase in adaptivity (Section 2.1). At the same time, the optimal value (i.e., spatial mapping uncertainty) decreases with increasing adaptivity. Note that 1-MASP corresponds to MASP in (2) while $n$-MASP is of the same form as the non-adaptive exploration problem in (1).

The computational efficiency of $j$-MASP does not improve with decreasing adaptivity (i.e., increasing $j$): for a fixed number of new locations (i.e., $kn$) to be sampled, the required number of stages decreases with decreasing adaptivity. But, it is associated with increasing dimensionality of the action space under each minimum and also, of the probability distribution for each expectation. If these expectations have to be evaluated numerically, the number of value function evaluations required for each expectation has to grow exponentially with the number of locations sampled per stage for the numerical approximation to be effective. Since the action space under each minimum also grows exponentially with the number of robot team actions per stage, there is no computational gain by decreasing the adaptivity.

## 2.4 Alternative Formulation

In this subsection, we provide an alternative formulation to MASP in (2) that lends itself to a different interpretation. More importantly, this reformulation can be subject to convex analysis (Section 2.6), which allows monotone-bounding approximation of MASP to be developed (Section 3.1).

The reformulated MASP comprises the value functions

$$\begin{aligned} U_i(d_{ki}) &= \max_{\mathbf{a}_i \in \mathcal{A}(\mathbf{x}_{d_i^k})} R(\mathbf{x}_{D_{i+1}^k}, d_{ki}) + \mathbb{E}[U_{i+1}(d_{ki}, D_{i+1}^k) \mid d_{ki}] \\ U_t(d_{kt}) &= \max_{\mathbf{a}_t \in \mathcal{A}(\mathbf{x}_{d_t^k})} R(\mathbf{x}_{D_{t+1}^k}, d_{kt}) \end{aligned} \tag{5}$$

for $i = 0, \ldots, t-1$ with $t = n-1$ and the reward functions

$$R(\mathbf{x}_{D_{i+1}^k}, d_{ki}) = \sum_{x_0 \in \mathcal{X}} \text{var}[\mu_{Z_{x_0}|d_{ki}, D_{i+1}^k} \mid d_{ki}] \tag{6}$$

where $\mu_{Z_{x_0}|d_{ki}, D_{i+1}^k} \stackrel{\text{def}}{=} \mathbb{E}[Z_{x_0}|d_{ki}, D_{i+1}^k]$. An analog to Theorem 2.1 can be derived for the reformulated MASP except that the optimal value increases, rather than decreases, with increasing adaptivity.

THEOREM 2.2. *The value functions of MASPs in (2) and (5) are related by*

$$V_i(d_{ki}) = \sum_{x_0 \in \mathcal{X}} \sigma^2_{Z_{x_0}|d_{ki}} \ - U_i(d_{ki}) \tag{7}$$

*for $i = 0, \ldots, n-1$ and their respective optimal action policies coincide.*

Theorem 2.2 can be generalized to cater to $j$-MASPs.

The reformulated MASP in (5) is interpreted differently from the original MASP in (2): from the well-known variance decomposition formula

$$\sigma^2_{Z_{x_0}|d_{ki}} = \mathbb{E}[\sigma^2_{Z_{x_0}|d_{ki}, D_{i+1}^k} \mid d_{ki}] + \text{var}[\mu_{Z_{x_0}|d_{ki}, D_{i+1}^k} \mid d_{ki}],$$

the $\text{var}[\mu_{Z_{x_0}|d_{ki},D^k_{i+1}}|d_{ki}]$ term measures the reduction in uncertainty at location $x_0$ from the prior variance $\sigma^2_{Z_{x_0}|d_{ki}}$ to the expected posterior variance $\mathbb{E}[\sigma^2_{Z_{x_0}|d_{ki},D^k_{i+1}}|d_{ki}]$ by sampling the new locations $\mathbf{x}_{D^k_{i+1}}$. So, by exploring new locations $\mathbf{x}_{D^k_{i+1}}$ at every stage that achieve greater reduction in uncertainty over all locations in $\mathcal{X}$ (i.e., maximizing rewards in (5)), we remove the largest possible amount of uncertainty from the initial spatial mapping uncertainty (i.e., $\sum_{x_0\in\mathcal{X}}\sigma^2_{Z_{x_0}|d_{ki}}$ in (7)). This is in contrast to the original MASP in (2) whereby the cost to be minimized appears only in the last stage (i.e., final spatial mapping uncertainty $\sum_{x_0\in\mathcal{X}}\sigma^2_{Z_{x_0}|d_{kn}}$).

In the next two subsections, we will show how the reformulated MASP in (5) can be applied to the Gaussian and log-Gaussian processes. In particular, we will analyze whether MASP is adaptive for these processes and the convex properties of MASP for the log-Gaussian process.

## 2.5 MASP for Gaussian Process (GP)

Let $\{Z_x\}_{x\in\mathcal{X}}$ denote a GP defined on the domain $\mathcal{X}$, that is, the joint distribution over any finite subset of $\{Z_x\}_{x\in\mathcal{X}}$ is Gaussian. The GP can be completely specified by its mean function $\mu_{Z_x} \stackrel{\text{def}}{=} \mathbb{E}[Z_x]$ and covariance function $\sigma_{Z_xZ_y} \stackrel{\text{def}}{=} \text{cov}[Z_x, Z_y]$ for $x, y \in \mathcal{X}$. In this paper, it is assumed that the mean function and covariance structure of $Z_x$ are known. Given the previously sampled data $d_{kn}$, the distribution of $Z_{x_0}$ is a Gaussian with the conditional mean and variance

$$\mu_{Z_{x_0}|d_{kn}} = \mu_{Z_{x_0}} + \Sigma_{x_0\mathbf{x}_{d_{kn}}}\Sigma^{-1}_{\mathbf{x}_{d_{kn}}\mathbf{x}_{d_{kn}}}\{\mathbf{z}^\top_{d_{kn}} - \boldsymbol{\mu}_{\mathbf{Z}_{d_{kn}}}\} \quad (8)$$

$$\sigma^2_{Z_{x_0}|d_{kn}} = \sigma_{Z_{x_0}Z_{x_0}} - \Sigma_{x_0\mathbf{x}_{d_{kn}}}\Sigma^{-1}_{\mathbf{x}_{d_{kn}}\mathbf{x}_{d_{kn}}}\Sigma_{\mathbf{x}_{d_{kn}}x_0} \quad (9)$$

where $\boldsymbol{\mu}_{\mathbf{Z}_{d_{kn}}}$ is a column vector with mean components $\mu_{Z_{x_i}}$ for $i = 1, \ldots, m + kn$, $\Sigma_{x_0\mathbf{x}_{d_{kn}}}$ is a covariance vector with components $\sigma_{Z_{x_0}Z_{x_i}}$ for $i = 1, \ldots, m + kn$, $\Sigma_{\mathbf{x}_{d_{kn}}x_0}$ is the transpose of $\Sigma_{x_0\mathbf{x}_{d_{kn}}}$, and $\Sigma_{\mathbf{x}_{d_{kn}}\mathbf{x}_{d_{kn}}}$ is a covariance matrix with components $\sigma_{Z_{x_i}Z_{x_j}}$ for $i, j = 1, \ldots, m + kn$.

For GP, MASP can be reduced to be non-adaptive. This is a direct consequence of the following lemma:

LEMMA 2.3. *$R(\mathbf{x}_{D^k_{i+1}}, d_{ki})$ in (6) is independent of $\mathbf{z}_{d_{kt}}$ for $i = 0, \ldots, t$.*

The next theorem follows from Lemma 2.3 and (5):

THEOREM 2.4. *$U_i(d_{ki})$ and $\boldsymbol{\pi}^*_i(d_{ki})$ are independent of $\mathbf{z}_{d_{kt}}$ for $i = 0, \ldots, t$.*

Hence, the selection of new sampling locations $\mathbf{x}_{D^k_{i+1}}$ is independent of $\mathbf{z}_{d_{kt}}$ for $i = 0, \ldots, t$. As a result, MASP for GP can be reduced to a deterministic planning problem

$$U_0(d_0) = \max_{\mathbf{a}_0,\ldots,\mathbf{a}_t} \sum_{i=0}^{t} R(\mathbf{x}_{D^k_{i+1}}, d_{ki}) \ , \quad (10)$$

which aims to provide sufficient coverage for mapping the environmental field accurately. However, it does not account for maximization of sampling at hotspots (Lemma 2.3).

## 2.6 MASP for log-Gaussian Process ($\ell$GP)

Let $\{Y_x\}_{x\in\mathcal{X}}$ denote a $\ell$GP defined on the domain $\mathcal{X}$. That is, if we let $Z_x = \log Y_x$, then $\{Z_x\}_{x\in\mathcal{X}}$ is a GP (Section 2.5). So, $Y_x = \exp\{Z_x\}$ and $\ell$GP has the mean function $\mu_{Y_x} \stackrel{\text{def}}{=} \mathbb{E}[Y_x] = \exp\{\mu_{Z_x} + \sigma_{Z_xZ_x}/2\}$ and covariance function $\sigma_{Y_xY_y} \stackrel{\text{def}}{=} \text{cov}[Y_x, Y_y] = \mu_{Y_x}\mu_{Y_y}(\exp\{\sigma_{Z_xZ_y}\} - 1)$ for $x, y \in \mathcal{X}$. From Section 2.5, we know that the distribution of $Z_{x_0}$ given $d_{kn}$ is Gaussian. Since the transformation from $\mathbf{z}_{d_{kn}}$ to $\mathbf{y}_{d_{kn}}$ is invertible, the distribution of $Y_{x_0}$ given $d_{kn}$ is log-Gaussian with the conditional mean and variance:

$$\mu_{Y_{x_0}|d_{kn}} = \exp\{\mu_{Z_{x_0}|d_{kn}} + \sigma^2_{Z_{x_0}|d_{kn}}/2\} \quad (11)$$

$$\sigma^2_{Y_{x_0}|d_{kn}} = \mu^2_{Y_{x_0}|d_{kn}}(\exp\{\sigma^2_{Z_{x_0}|d_{kn}}\} - 1) \quad (12)$$

where $\mu_{Z_{x_0}|d_{kn}}$ and $\sigma^2_{Z_{x_0}|d_{kn}}$ are determined using (8) and (9) respectively.

For $\ell$GP, MASP is adaptive. This is a direct consequence of the following lemma:

LEMMA 2.5. *$R(\mathbf{x}_{D^k_{i+1}}, d_{ki})$ in (6) depends on $d_{ki}$ for $i = 0, \ldots, t$.*

The next theorem follows from Lemma 2.5 and (5):

THEOREM 2.6. *$U_i(d_{ki})$ and $\boldsymbol{\pi}^*_i(d_{ki})$ depend on $d_{ki}$ for $i = 0, \ldots, t$.*

Hence, the selection of new sampling locations $\mathbf{x}_{D^k_{i+1}}$ depends on the previously sampled data $d_{ki}$ for $i = 0, \ldots, t$.

Besides providing coverage to learn an accurate spatial mapping, MASP in (5) for $\ell$GP also maximizes hotspot sampling: it can be shown that a large reward $\text{var}[\mu_{Y_{x_0}|d_{ki},D^k_{i+1}}|d_{ki}]$ in (6) is associated with a high expected quantity $\mu_{Y_{x_0}|d_{ki}}$. So, if $x_0$ is one of the new sampling locations in $\mathbf{x}_{D^k_{i+1}}$, MASP in (5) tends to select location $x_0$ with a large reward $\text{var}[\mu_{Y_{x_0}|d_{ki},D^k_{i+1}}|d_{ki}]$. Consequently, $x_0$ has a high expected quantity, thus maximizing sampling of hotspots (i.e., areas of high measured quantities).

The value functions of MASP in (5) may not be convex in the sampled quantities $\mathbf{y}_d$ of the input data $d$. However, MASP can be transformed to be convex by a change of variables $Z_x = \log Y_x, x \in \mathcal{X}$ as shown below:

LEMMA 2.7. *$R(\mathbf{x}_{D^k_{i+1}}, d_{ki})$ in (6) is convex in $\mathbf{z}_{d_{ki}}$ (i.e., $(\log y_{x_1}, \ldots, \log y_{x_{m+ki}})$) for $i = 0, \ldots, t$.*

The next theorem follows from Lemma 2.7 and (5):

THEOREM 2.8. *$U_i(d_{ki})$ is convex in $\mathbf{z}_{d_{ki}}$ (i.e., $(\log y_{x_1}, \ldots, \log y_{x_{m+ki}})$) for $i = 0, \ldots, t$.*

# 3. VALUE-FUNCTION APPROXIMATIONS

The solution technique presented in this section focuses on tackling the strictly adaptive MASP in (5) (Section 2.1), which implies only one new location should be selected at each stage. The value functions can then be simplified in the following two ways resulting in (13): (a) rather than choose a joint action $\mathbf{a}_i \in \mathcal{A}(.)$ to move all robots simultaneously in each stage, only one robot should be chosen at each stage to sample a new location while the rest of the robots stay put. This tradeoff between simultaneous actions and strict adaptivity results in a reduced set $\mathcal{A}'(.)$ of joint actions for the robot team that grows linearly, rather than exponentially (Section 2.3), with the number of robots; (b) consequently, the unobserved data $D^k_{i+1}$ can be reduced to a single pair $\langle x', Z_{x'}\rangle$ corresponding to the new location $x'$ to be sampled

by the chosen robot. Since the other unselected robots are stationary in that stage, the remaining pairs in $D_{i+1}^k$ correspond to locations selected in the previous stages and can be found in the known data $d_i$. The probability distribution for the conditional expectation in MASP can therefore be simplified to a uni-variate $Z_{x'}$, which reduces the computational burden of solving the problem numerically (Section 2.3).

$$\begin{aligned} U_i(d_i) &= \max_{\mathbf{a}_i \in \mathcal{A}'(\mathbf{x}_i)} R(x', d_i) + \mathbb{E}[U_{i+1}(d_i, \langle x', Z_{x'}\rangle)|d_i] \\ U_t(d_t) &= \max_{\mathbf{a}_t \in \mathcal{A}'(\mathbf{x}_t)} R(x', d_t) \end{aligned} \tag{13}$$

for $i = 0, \ldots, t-1$ where $\mathbf{x}_i$ is a vector of the $k$ robot locations denoting the current state of the robot team that can be derived from the sampled locations $\mathbf{x}_{d_i}$, and $\mathbf{x}_{i+1}$ is the next state produced by the deterministic transition function $T(\mathbf{a}_i, \mathbf{x}_i)$. Note that $x'$ is the component in $\mathbf{x}_{i+1}$ with the same index as the non-zero component in $\mathbf{x}_{i+1} - \mathbf{x}_i$.

Since the random variable $Z_{x'}$ is continuous, an exact solution to the above MASP will not be computationally feasible if the conditional expectation is evaluated by computing $U_{i+1}(d_i, \langle x', Z_{x'}\rangle)$ infinitely often over the support of $Z_{x'}$. For MASP with $t = 1$ in (13), it can be shown that the conditional expectation can be evaluated in closed form, which makes the problem computationally feasible. At this moment, we are not aware of any computationally feasible methods to solve MASP with $t > 1$ exactly. Hence, we will resort to approximating MASP as described below. For ease of exposition, we will revert to using the $Z_x$ variable for $\ell$GP (i.e., by transforming $Z_x = \log Y_x$) in the rest of this paper.

The difficulty in solving the multi-stage MASP lies in evaluating the conditional expectation with respect to the continuous state variable $Z_{x'}$. This intricate issue of handling continuous states is faced by the following related stochastic decision-theoretic planning problems, which have resolved it by constructing approximate problems:

**Markov decision processes (MDPs)** The traditional approach of generalizing to continuous states in an MDP is to approximate the value function with a parameterized model; the resulting solution is usually hard to analyze and may diverge. To make the problem computationally feasible to solve, recent approaches such as the time-dependent [8] and factored MDPs [6] approximate it by constraining the transition, reward, and value functions to certain function families. However, time-dependent MDPs suffer from an exponential blow-up with an increasing number of stages while factored MDPs induce infinitely many constraints in their linear programming formulation.

In contrast, MASP adopts a more complex but realistic non-Markov structure; the transition function is conditioned on the entire history of actions and continuous states. More importantly, by assuming the reward and value functions to be convex (Section 2.6), piecewise-linear functions can be constructed to monotonically bound and approximate the value function (Section 3.1). Note that the form of the transition function is not restricted.

**Non-Markov problems** Bayes sequential design problems [11] and stochastic programs [4, 15] can be modeled as non-Markov DP problems. In contrast to MASP, they have a simple structure: (a) their transition functions do not depend on past actions, (b) for Bayes sequential design, the entire history of continuous states can be reduced to a summary statistic, and (c) for stochastic programs, the reward function is often assumed to be linear in the action variable [4]. To make them computationally feasible to solve, the conditional expectation is approximated using Monte-Carlo sampling for both problems [11, 15] and bounding methods [4] for stochastic programs. The latter technique further assumes the value function to be linear or convex in the continuous state and action variables. The resulting approximate problems suffer from an exponential blow-up. Our bounding approximation technique (Section 3.1) utilizes the results on generalized Jensen bounds for convex functions [5] from the field of stochastic programming.

## 3.1 Approximate MASP

In this section, we will formulate the approximate MASP (aMASP) whose optimal value lower-bounds that of MASP in (13). To obtain aMASP and its corresponding bound, the following result is required, which utilizes Jensen's inequality to lower-bound the expectation of a convex function:

THEOREM 3.1 ([5]). *Let $W(\xi)$ be a convex function of $\xi$ with the support $[a, b]$ that is subdivided at arbitrary points $b_0, \ldots, b_\nu$ (i.e., $a := b_0 < b_1 < \ldots < b_\nu =: b$). Let the $\nu$-fold generalized Jensen bound be denoted by*

$$J_\nu \stackrel{\text{def}}{=} \sum_{j=1}^{\nu} \alpha_j W(\beta_j), \quad \nu = 1, 2, \ldots, \tag{14}$$

*where*

$$\alpha_j \stackrel{\text{def}}{=} \int_{b_{j-1}}^{b_j} f(\xi)d\xi, \quad \beta_j \stackrel{\text{def}}{=} \frac{1}{\alpha_j} \int_{b_{j-1}}^{b_j} \xi f(\xi)d\xi, \quad j = 1, \ldots, \nu\ .$$

*If the partition corresponding to $k+1$ is at least as fine as that corresponding to $k$ for $k = 1, \ldots, \nu-1$, $J_1 \leq \ldots \leq J_\nu \leq \mathbb{E}[W(\xi)]$.*

Let the support of $Z_{x'}$ given the sampled data $d_i$ be $S_{x'i}^\nu = [a, b]$ that is partitioned into $\nu$ non-empty, disjoint intervals $S_{x'ij}^\nu = [b_{j-1}, b_j]$ for $j = 1, \ldots, \nu$. Using Theorem 3.1, aMASP can be derived from (13) with the structure:

$$\begin{aligned} \underline{U}_i^\nu(d_i) &= \max_{\mathbf{a}_i \in \mathcal{A}'(\mathbf{x}_i)} R(x', d_i) + \sum_{j=1}^{\nu} p_{x'ij} \underline{U}_{i+1}^\nu(d_i, \langle x', z_{x'ij}\rangle) \\ \underline{U}_t^\nu(d_t) &= \max_{\mathbf{a}_t \in \mathcal{A}'(\mathbf{x}_t)} R(x', d_t) \end{aligned} \tag{15}$$

for $i = 0, \ldots, t-1$ where $p_{x'ij} \stackrel{\text{def}}{=} P(z_{x'} \in S_{x'ij}^\nu | d_i)$, and $z_{x'ij} \stackrel{\text{def}}{=} \mu_{Z_{x'}|d_i, S_{x'ij}^\nu}$, which is the expectation of $Z_{x'}$ conditioned on $d_i$ and $z_{x'} \in S_{x'ij}^\nu$. Note that the parameters of aMASP correspond to that of the Jensen bound in (14). More importantly, the structure of aMASP in (15) can be viewed as approximating the continuous state variable $Z_{x'}$ in (13) using a discrete one with a distribution at points $z_{x'ij}$ of probability $p_{x'ij} > 0$ for $j = 1, \ldots, \nu$ where $\sum_{j=1}^{\nu} p_{x'ij} = 1$.

The optimal action policy $\boldsymbol{\pi}^{\nu *} = \langle \boldsymbol{\pi}_0^{\nu *}(d_0), \ldots, \boldsymbol{\pi}_t^{\nu *}(d_t)\rangle$ is defined in a similar manner as that of (3). Different from (3), the additional quantities $[\langle x_{m+1}, z_{x_{m+1}}\rangle, \ldots, \langle x_{m+i}, z_{x_{m+i}}\rangle]$ observed during exploration are expected to be realized from discrete, rather than continuous, distributions as explained above. If replanning is not allowed during exploration, then we have to choose the most appropriate action rule to apply to the observed continuous quantities. This is resolved in a forward stagewise manner: when $\langle x_{m+1}, z_{x_{m+1}}\rangle$ is sampled during exploration, the next action $\boldsymbol{\pi}_1^{\nu *}(d_0, \langle x_{m+1}, z_{x_{m+1}0j_0}\rangle)$

is selected such that $j_0 = \arg\min_j |z_{x_{m+1}} - z_{x_{m+1}0j}|$. When $\langle x_{m+2}, z_{x_{m+2}}\rangle$ is sampled, the next action $\boldsymbol{\pi}_2^{\nu*}(d_0, \langle x_{m+1}, z_{x_{m+1}0j_0}\rangle, \langle x_{m+2}, z_{x_{m+2}1j_1}\rangle)$ is selected such that $j_1 = \arg\min_j |z_{x_{m+2}} - z_{x_{m+2}1j_1}|$. This goes on for $i = 3, \ldots, t$ such that when $\langle x_{m+i}, z_{x_{m+i}}\rangle$ is sampled, the next action $\boldsymbol{\pi}_i^{\nu*}(d_0, \langle x_{m+1}, z_{x_{m+1}0j_0}\rangle, \ldots, \langle x_{m+i}, z_{x_{m+i}(i-1)j_{i-1}}\rangle)$ is selected with $j_{i-1} = \arg\min_j |z_{x_{m+i}} - z_{x_{m+i}(i-1)j_{i-1}}|$.

To prove the monotone bounds, the value functions of MASP are required to be convex, which have been shown for $\ell$GP (Section 2.6). Together with Theorem 3.1 and the following lemma, we can derive the bounds in Theorem 3.3:

LEMMA 3.2. *$\underline{U}_i^{\nu}(d_i)$ is convex in $\mathbf{z}_{d_i}$ for $i = 0, \ldots, t$.*

THEOREM 3.3. *If $S_{x'i}^{\nu+1}$ is obtained by splitting one of the intervals in $S_{x'i}^{\nu}$, $\underline{U}_i^{\nu}(d_i) \leq \underline{U}_i^{\nu+1}(d_i) \leq U_i(d_i)$ for $i = 0, \ldots, t$.*

More importantly, Theorem 3.3 indicates that aMASP yields a lower bound $\underline{U}_0^{\nu}(d_0)$ to the optimal value $U_0(d_0)$ of MASP in (13). Furthermore, by refining the partition (i.e., increasing $\nu$), the bounds can be improved; the optimal value of aMASP monotonically increases to that of MASP. However, this increases the computational burden of solving aMASP. The next corollary is a direct result of Theorems 2.2 and 3.3:

COROLLARY 3.4. *Let $\overline{V}_0^{\nu}(d_0) = \sum_{x_0 \in \mathcal{X}} \sigma^2_{Z_{x_0}|d_0} - \underline{U}_0^{\nu}(d_0)$. Then, $V_0(d_0) \leq \overline{V}_0^{\nu+1}(d_0) \leq \overline{V}_0^{\nu}(d_0)$.*

THEOREM 3.5. *Let*

$$Q_i(\boldsymbol{\pi}, d_i) = R(x'_{\boldsymbol{\pi}_i(d_i)}, d_i) + \mathbb{E}[Q_{i+1}(\boldsymbol{\pi}, [d_i, \langle x'_{\boldsymbol{\pi}_i(d_i)}, Z_{x'_{\boldsymbol{\pi}_i(d_i)}}\rangle])|d_i]$$
$$Q_t(\boldsymbol{\pi}, d_t) = R(x'_{\boldsymbol{\pi}_t(d_t)}, d_t)$$

*for $i = 0, \ldots, t-1$ where $x'_{\boldsymbol{\pi}_i(d_i)}$ is the new location to be sampled by taking the current action $\boldsymbol{\pi}_i(d_i)$. If $\boldsymbol{\pi}^{\nu*}$ is the optimal action policy for aMASP, $\underline{U}_i^{\nu}(d_i) \leq Q_i(\boldsymbol{\pi}^{\nu*}, d_i) \leq U_i(d_i)$ for $i = 0, \ldots, t$.*

Theorem 3.5 indicates that if the optimal action policy $\boldsymbol{\pi}^{\nu*}$ derived by solving aMASP is used in the original MASP, the lower bound is improved (i.e., the policy $\boldsymbol{\pi}^{\nu*}$ is guaranteed to achieve no worse than $\underline{U}_i^{\nu}(d_i)$ for MASP).

# 4. REAL-TIME DYNAMIC PROGRAMMING

For our bounding approximation scheme, the state size grows exponentially with the number of stages. This is due to the nature of DP-based problems, which takes into account all possible states. To alleviate this computational difficulty, we propose anytime algorithms based on aMASP, one of which can guarantee its policy quality for the original MASP in real time.

Our proposed anytime algorithms are adapted from the *Real-Time Dynamic Programming* (RTDP) [1] technique, which is a well-known heuristic search algorithm for discrete-state MDPs. RTDP essentially simulates greedy exploration paths through a large state space. This results in the following desirable properties: (a) the search is focused, that is, it does not have to evaluate the entire state space to obtain the optimal policy, and (b) it has a good anytime behavior, that is, it produces a good policy fast and this policy improves over time. The disadvantage of RTDP is its slow convergence due to the focused search.

A non-trivial issue arises with generalizing RTDP to handle the non-Markov structure of aMASP: the state space of MDP is often assumed to be tractable. Based on this assumption, RTDP has been enhanced in [2, 3] with additional procedures to improve convergence, which require time complexity linear in the state size. More importantly, improvements of RTDP [2, 3, 10, 19] emphasize the use of informed heuristic bounds, which are preprocessed with time complexity linear in the state size. This is clearly unacceptable for our anytime algorithms since the state size of aMASP grows exponentially with the number of stages. In the next section, we will derive informed heuristic bounds that are computationally efficient.

## 4.1 Preprocessing of Heuristic Bounds

The greedy exploration in RTDP is guided by heuristic bounds, which are used to prune unnecessary, bad searches of the state space while still guaranteeing policy optimality. In particular, when the initial bounds are more informed or tighter (as opposed to non-informed, loose bounds used in [1]), the anytime and convergence performance can be improved [2, 3, 10]. However, this makes the preprocessing of bounds more computationally expensive as described earlier.

To obtain computationally efficient informed lower bounds, we can relax aMASP with $\nu = 1$ by choosing the best action to maximize the immediate reward at each stage:

$$\begin{array}{ll} \underline{H}_i(d_i) = & R(x'^*, d_i) + \underline{H}_{i+1}(d_i, \langle x'^*, \mu_{Z_{x'^*}|d_i}\rangle) \\ \underline{H}_t(d_t) = & \max\limits_{\mathbf{a}_t \in \mathcal{A}'(\mathbf{x}_t)} R(x', d_t) \end{array} \quad (16)$$

for $i = 0, \ldots, t-1$ where $R(x'^*, d_i) = \max_{\mathbf{a}_i \in \mathcal{A}'(\mathbf{x}_i)} R(x', d_i)$. It is easy to see that $\underline{H}_i(d_i) \leq \underline{U}_i^1(d_i)$ and therefore lower-bounds $\underline{U}_i^{\nu}(d_i)$ of aMASP (Theorem 3.3). It can be shown that

$$\underline{H}_i(d_i) \leq \max_{\mathbf{a}_i \in \mathcal{A}'(\mathbf{x}_i)} R(x', d_i) + \sum_{j=1}^{\nu} p_{x'ij} \underline{H}_{i+1}(d_i, \langle x', z_{x'ij}\rangle) \ .$$

Then, we say that the lower heuristic bound is monotonic. However, the state space only grows linearly with the number $t$ of stages for computing this lower bound.

To derive computationally efficient informed upper bounds, Theorem 2.2 can be exploited to give

$$\overline{H}_i(d_i) = \sum_{x_0 \in \mathcal{X}} \sigma^2_{Y_{x_0}|d_i}, \quad \overline{H}_t(d_t) = \max_{\mathbf{a}_t \in \mathcal{A}'(\mathbf{x}_t)} R(x', d_t) \quad (17)$$

for $i = 0, \ldots, t-1$. From Theorem 2.2, $\overline{H}_i(d_i)$ upper-bounds $U_i(d_i)$ since $V_i(d_i) \geq 0$. Therefore, it upper-bounds $\underline{U}_i^{\nu}(d_i)$ of aMASP (Theorem 3.3). This upper bound can be computed with time complexity constant in the number of stages. It can be shown that

$$\overline{H}_i(d_i) \geq \max_{\mathbf{a}_i \in \mathcal{A}'(\mathbf{x}_i)} R(x', d_i) + \sum_{j=1}^{\nu} p_{x'ij} \overline{H}_{i+1}(d_i, \langle x', z_{x'ij}\rangle) \ .$$

Then, we say that the upper heuristic bound is monotonic.

## 4.2 Anytime Algorithms

The first anytime algorithm (Algorithm 1) is adapted directly from RTDP. To elaborate, each simulated exploration path involves an alternating selection of actions and their corresponding outcomes until the last stage is reached; each action is selected based on the upper bound (line 3) and the corresponding next state/outcome to explore is chosen based on the discrete distribution of aMASP (line 4). Then,

the algorithm backtracks up the path to update the upper heuristic bounds (lines 8-10) using $\max_{\mathbf{a}_i} \overline{Q}_i(\mathbf{a}_i, d_i)$ where

$$\overline{Q}_i(\mathbf{a}_i, d_i) \stackrel{\text{def}}{=} R(x', d_i) + \sum_{j=1}^{\nu} p_{x'ij} \overline{U}_{i+1}(d_i, \langle x', z_{x'ij} \rangle) \ .$$

We assume that whenever a new state is encountered, it is initialized with the upper bound derived in Section 4.1. When an action policy is requested at any time during the algorithm's execution, we provide the greedy policy induced by the upper bound. But, its quality is not guaranteed.

RTDP($d_0, t$):
  **while** true **do**
    SIMULATED-PATH($d_0, t$)

SIMULATED-PATH($d_0, t$):
1: $i \leftarrow 0$
2: **while** $i < t$ **do**
3: $\quad \mathbf{a} \leftarrow \arg\max_{\mathbf{a}_i} \overline{Q}_i(\mathbf{a}_i, d_i)$
4: $\quad z \leftarrow$ sample from distribution at points $z_{x'_{\mathbf{a}}ij}$ of probability $p_{x'_{\mathbf{a}}ij}$ for $j = 1, \ldots, \nu$
5: $\quad d_{i+1} \leftarrow d_i, \langle x'_{\mathbf{a}}, z \rangle$
6: $\quad i \leftarrow i + 1$
7: $\overline{U}_i(d_i) \leftarrow \max_{\mathbf{a}_i} R(x', d_i)$
8: **while** $i > 0$ **do**
9: $\quad i \leftarrow i - 1$
10: $\quad \overline{U}_i(d_i) \leftarrow \max_{\mathbf{a}_i} \overline{Q}_i(\mathbf{a}_i, d_i)$

**Algorithm 1:** RTDP.

The second anytime algorithm (Algorithm 2) is inspired by the uncertainty-based RTDP (URTDP) techniques [10, 19]. Different from Algorithm 1, it maintains both lower and upper bounds for each encountered state, which are used to derive the uncertainty of its corresponding optimal value function. The algorithm exploits them to guide future searches in a more informed manner; it explores the next state/outcome with the greatest amount of uncertainty (lines 4-5). Besides updating the upper bounds, the lower heuristic bounds are also updated during backtracking via $\max_{\mathbf{a}_i} \underline{Q}_i(\mathbf{a}_i, d_i)$ (lines 10-13) where

$$\underline{Q}_i(\mathbf{a}_i, d_i) \stackrel{\text{def}}{=} R(x', d_i) + \sum_{j=1}^{\nu} p_{x'ij} \underline{U}_{i+1}(d_i, \langle x', z_{x'ij} \rangle) \ .$$

When an action policy is requested, we provide the greedy policy induced by the lower bound. The quality of this policy has a similar guarantee to Theorem 3.5 whereby the anytime algorithm for solving aMASP provides a greedy policy that achieves no worse than $\underline{U}_0(d_0)$ for using in MASP.

## 5. EXPERIMENTS AND DISCUSSION

This section presents empirical evaluations of aMASP on a real-world dataset, that is, the June 2006 monthly composite plankton density data of Chesapeake Bay from NOAA CoastWatch bounded within the latitude 38.481-38.591N and longitude 76.487-76.335W. The bay area (Fig. 1) is discretized into a $14 \times 12$ grid of sampling units. Each unit $x$ is associated with a plankton density $Y_x$ measured in chlorophyll-a (chl-a). The exploration region (i.e., sea) comprises $|\mathcal{X}| = 148$ such units enclosed by the dark blue boundary (Fig. 1). A fleet of two robotic boats is tasked to explore 18 sampling units in this bay area with their starting locations indicated in Fig. 1; each robot is constrained to sample 9 adjacent units in its path including the unit it starts in. If only one robot is used for exploration, it is placed in the top starting unit (Fig. 1) and has to sample all 18 units. The actions of each robot are restricted to moving to the front, left, or right unit. Instead of assuming the mean function and covariance structure of GP and $\ell$GP to be known, we use the data of 20 randomly selected units to learn their hyperparameters through maximum likelihood estimation [14]. So, the known data $d_0$ comprises the randomly selected units and the starting units of the robots.

URTDP($d_0, t$):
  **while** $\overline{U}_0(d_0) - \underline{U}_0(d_0) > \alpha$ **do**
    SIMULATED-PATH($d_0, t$)

SIMULATED-PATH($d_0, t$):
1: $i \leftarrow 0$
2: **while** $i < t$ **do**
3: $\quad \mathbf{a} \leftarrow \arg\max_{\mathbf{a}_i} \overline{Q}_i(\mathbf{a}_i, d_i)$
4: $\quad \forall j, \ \Xi(j) \leftarrow p_{x'_{\mathbf{a}}ij} \ \{\overline{U}_{i+1}(d_i, \langle x'_{\mathbf{a}}, z_{x'_{\mathbf{a}}ij} \rangle) - \underline{U}_{i+1}(d_i, \langle x'_{\mathbf{a}}, z_{x'_{\mathbf{a}}ij} \rangle)\}$
5: $\quad z \leftarrow$ sample from distribution at points $z_{x'_{\mathbf{a}}ij}$ of probability $\Xi(j) / \sum_k \Xi(k)$ for $j = 1, \ldots, \nu$
6: $\quad d_{i+1} \leftarrow d_i, \langle x'_{\mathbf{a}}, z \rangle$
7: $\quad i \leftarrow i + 1$
8: $\overline{U}_i(d_i) \leftarrow \max_{\mathbf{a}_i} R(x', d_i)$
9: $\underline{U}_i(d_i) \leftarrow \max_{\mathbf{a}_i} R(x', d_i)$
10: **while** $i > 0$ **do**
11: $\quad i \leftarrow i - 1$
12: $\quad \overline{U}_i(d_i) \leftarrow \max_{\mathbf{a}_i} \overline{Q}_i(\mathbf{a}_i, d_i)$
13: $\quad \underline{U}_i(d_i) \leftarrow \max_{\mathbf{a}_i} \underline{Q}_i(\mathbf{a}_i, d_i)$

**Algorithm 2:** URTDP ($\alpha$ is user-specified bound).

The policy performance of strictly adaptive aMASP is compared to that of the state-of-the-art exploration strategies, namely, the greedy and non-adaptive strategies. The greedy strategies are applied to sampling GP and $\ell$GP; a greedy strategy repeatedly chooses a reward-maximizing action (i.e., by repeatedly solving MASP with $t = 0$ in (13)) to obtain the robot paths. The non-adaptive strategy for GP corresponds to the deterministic planning problem in (10). Similar to aMASP, its state size grows exponentially with the number of stages. Therefore, it is approximated by a deterministic version of RTDP called LRTA$^*$.

Two performance metrics are used to evaluate the policies of the above exploration strategies: (a) Mean-Squared Relative Error (MSRE) $|\mathcal{X}|^{-1} \sum_{x \in \mathcal{X}} \{(Y_x - \mu_{Y_x|d_t})/\bar{\mu}\}^2$ measures the spatial mapping uncertainty by using $\mu_{Y_x|d_t}$ in (11) to predict the plankton density field where $\bar{\mu} = |\mathcal{X}|^{-1} \sum_{x \in \mathcal{X}} Y_x$, and $t = 16$ (17) for the case of 2 (1) robots. A small MSRE implies lower uncertainty and thus, better wide-area coverage; (b) chl-a yield measures the amount of plankton sampled by the robot paths; a high plankton yield means greater sampling at hotspots.

Table 1 shows the results of various exploration strategies with different assumed models and robot team size. For the adaptive aMASPs and non-adaptive MASP, the results are obtained using the action policies derived after running 100000 simulated paths. The results show that the strategies for $\ell$GP obtain higher plankton yield than that for GP.

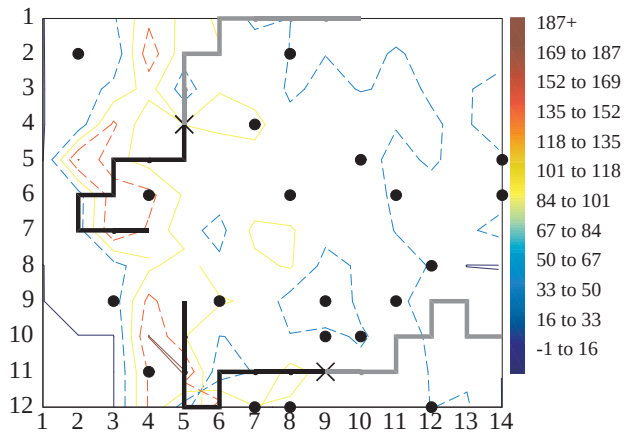


**Figure 1: Plankton density (chl-a) field of Chesapeake Bay: 20 units (black dots) are randomly selected as known data. The robots start at locations marked by '×'s. The black and gray robot paths are produced by adaptive aMASP for ℓGP and non-adaptive MASP for GP respectively.**

**Table 1: Performance comparison of robot exploration strategies: 1R and 2R denote 1 and 2 robots respectively.**

| | | MSRE | | chl-a yield | |
|---|---|---|---|---|---|
| Exploration strategy | Model | 1R | 2R | 1R | 2R |
| Adaptive aMASP/RTDP | ℓGP | 0.284 | 0.241 | 1660 | 1607 |
| Adaptive aMASP/URTDP | ℓGP | 0.250 | 0.197 | 1652 | 1815 |
| Greedy | ℓGP | 0.338 | 0.260 | 1840 | 1647 |
| Non-adaptive MASP | GP | 0.325 | 0.333 | 1165 | 1240 |
| Greedy | GP | 0.401 | 0.407 | 967 | 982 |

In particular, the adaptive aMASP with URTDP achieves lowest MSRE and very high plankton yield as compared to the non-adaptive and greedy strategies. Furthermore, it can be observed from Fig. 1 that the action policy of adaptive aMASP with URTDP moves the robots to sample the hotspots but that of non-adaptive MASP for GP does not. Therefore, the adaptive aMASP with URTDP is capable of performing superior wide-area coverage (lowest MSRE) and hotspot sampling (very high plankton yield).

## 6. CONCLUSIONS

This paper describes an adaptive multi-robot exploration strategy based on MASP that is novel in performing both wide-area coverage and hotspot sampling. A key feature of MASP is in covering the entire adaptivity spectrum; a theoretical analysis of MASP with varying adaptivity reveals that a more adaptive strategy reduces spatial mapping uncertainty. We demonstrate its applicability to sampling GP and ℓGP, which result in non-adaptive and adaptive exploration strategies respectively. We also show that MASP for ℓGP caters to both wide-area coverage and hotspot sampling while that for GP only achieves the former. Since it is non-trivial to solve MASP due to its continuous state components, it is approximated by discrete-state monotone-bounding aMASP. We provide a theoretical guarantee on the policy quality of aMASP for using in the original MASP. To alleviate the computational difficulty of solving aMASP for ℓGP, anytime algorithms are proposed based on aMASP: the URTDP algorithm can guarantee its policy quality for the original MASP in real time and is demonstrated empirically to achieve superior wide-area coverage and hotspot sampling as compared to state-of-the-art strategies.

## 7. REFERENCES


[1] A. Barto, S. Bradtke, and S. Singh. Learning to act using real-time dynamic programming. *Artif. Intell.*, 72(1-2):81–138, 1995.

[2] B. Bonet and H. Geffner. Faster heuristic search algorithms for planning with uncertainty and full feedback. In *Proc. IJCAI*, pages 1233–1238, 2003.

[3] B. Bonet and H. Geffner. Labeled RTDP: Improving the convergence of real-time dynamic programming. In *Proc. ICAPS*, pages 12–21, 2003.

[4] N. C. P. Edirisinghe. Bound-based approximations in multistage stochastic programming: Nonanticipativity aggregation. *Ann. Oper. Res.*, 85(1):103–127, 1999.

[5] C. C. Huang, W. T. Ziemba, and A. Ben-Tal. Bounds on the expectation of a convex function of a random variable: With applications to stochastic programming. *Oper. Res.*, 25(2):315–Ű325, 1977.

[6] B. Kveton, M. Hauskrecht, and C. Guestrin. Solving factored MDPs with hybrid state and action variables. *J. Artif. Intell. Res.*, 27:153Ű–201, 2006.

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

[8] L. Li and M. L. Littman. Lazy approximation for solving continuous finite-horizon MDPs. In *Proc. AAAI*, pages 1175–1180, 2005.

[9] K. H. Low, G. J. Gordon, J. M. Dolan, and P. Khosla. Adaptive sampling for multi-robot wide-area exploration. In *Proc. ICRA*, pages 755–760, 2007.

[10] H. B. McMahan, M. Likhachev, and G. J. Gordon. Bounded real-time dynamic programming: RTDP with monotone upper bounds and performance guarantees. In *Proc. ICML*, pages 569–576, 2005.

[11] P. Müller, D. A. Berry, A. P. Grieve, M. Smith, and M. Krams. Simulation-based sequential Bayesian design. *J. Statist. Plan. Infer.*, 137:3140–3150, 2007.

[12] D. O. Popa, M. F. Mysorewala, and F. L. Lewis. EKF-based adaptive sampling with mobile robotic sensor nodes. In *Proc. IROS*, pages 2451–2456, 2006.

[13] M. Rahimi, R. Pon, W. J. Kaiser, G. S. Sukhatme, D. Estrin, and M. Srivastava. Adaptive sampling for environmental robotics. In *Proc. ICRA*, pages 3536–3544, 2004.

[14] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[15] A. Shapiro. On complexity of multistage stochastic programs. *Oper. Res. Lett.*, 34(1):1–8, 2006.

[16] R. Sim and N. Roy. Global A-optimal robot exploration in SLAM. In *Proc. ICRA*, 2005.

[17] A. Singh, A. Krause, C. Guestrin, W. Kaiser, and M. Batalin. Efficient planning of informative paths for multiple robots. In *Proc. IJCAI*, 2007.

[18] A. Singh, R. Nowak, and P. Ramanathan. Active learning for adaptive mobile sensing networks. In *Proc. IPSN*, pages 60–68, 2006.

[19] T. Smith and R. Simmons. Focused real-time dynamic programming for MDPs: Squeezing more out of a heuristic. In *Proc. AAAI*, 2006.