Robotics: Science and Systems 2019
Freiburg im Breisgau, June 22-26, 2019

Pareto Monte Carlo Tree Search for Multi-Objective Informative Planning

Weizhe Chen, Lantao Liu

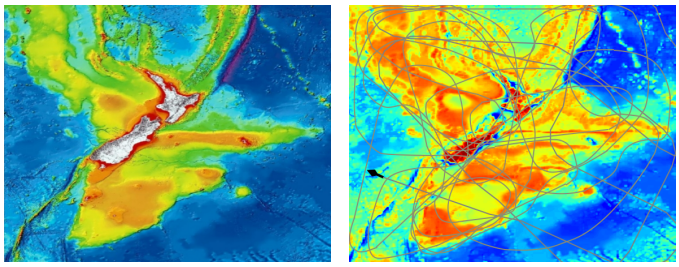
***Abstract*—In many environmental monitoring scenarios, the sampling robot needs to simultaneously explore the environment and exploit features of interest with limited time. We present an anytime multi-objective informative planning method called Pareto Monte Carlo tree search which allows the robot to handle potentially competing objectives such as exploration versus exploitation. The method produces optimized decision solutions for the robot based on its knowledge (estimation) of the environment state, leading to better adaptation to environmental dynamics. We provide algorithmic analysis on the critical tree node selection step and show that the number of times choosing sub-optimal nodes is logarithmically bounded and the search result converges to the optimal choices at a polynomial rate.**

I. INTRODUCTION

There is an increasing need that mobile robots are tasked to gather information from our environment. Navigating robots to collect samples with the largest amount of information is called informative planning [4, 21, 25]. The basic idea is to maximize information gain using information-theoretic methods, where the information gain (or *informativeness*) is calculated from the estimation uncertainty based on some prediction models such as the set of Gaussian systems. Informative planning is challenging due to the large and complex searching space. Such problems have been shown to be NP-hard [26] or PSPACE-hard [23] depending on the form of objective functions and the corresponding searching space.

Existing informative planning work is heavily built upon information-theoretic framework. The produced solutions tend to guide the robot to explore the uncertain or unknown areas and reduce the estimation uncertainty (e.g., the entropy or mutual information based efforts). Aside from the estimation uncertainty, there are many other estimation properties that we are interested in. For example, when monitoring some biological activity in a lake, scientists are interested in areas with high concentration – known as *hotspots* – of biological activity [20]. Similarly, plume tracking is useful for detection and description of oil spills, thermal vents, and harmful algal blooms [29]. In these tasks, the robot needs to explore the environment to discover hotspots first, and then visit different areas with differing effort (frequency) to obtain the most valuable samples and catch up with the (possibly) spatiotemporal environmental dynamics.

Since exploration and exploitation cannot be achieved simultaneously most of the time, different planning objectives may lead to distinct motion trajectories. The problem falls within the spectrum of multi-objective optimization in which the optima are not unique. In fact, continua of optimal solutions are possible. The goal is to find the Pareto optimal set, or so called Pareto front. The solutions in the Pareto optimal set cannot be improved for any objective without hurting other objectives. A straightforward solution to the multi-objective problem is to convert it into a single-objective problem by linear scalarization (weighted sum of objectives is one of the linear scalarization method). However, linear scalarization might be impossible, infeasible, or undesirable for the following reasons. First of all, objectives might be conflicting. Taking algal bloom monitoring as an example, high-concentration areas and the areas with high temporal variability (i.e., spreading dynamics) are both important. We expect the robot to visit these areas more frequently, but these two types of areas are not necessarily in the same direction. Furthermore, the weights among different objectives are hard to determine in the linear scalarization approaches. Last but not least, linear scalarization based approaches fail to discover the solutions in the non-convex regions of the Pareto front.



(a) Bathymetry (b) Example trajectory

Fig. 1. Illustrative multi-objective informative planning trajectory for modeling the Bathymetry of New Zealand's Exclusive Economic Zone. The goal is to explore the environment while visiting areas with high value and variability more frequently. (Source: NIWA)

This paper presents the following contributions:

- Different from existing informative planning approaches where information-seeking is the main objective, we propose a new generic planning method that optimizes over multiple (possibly) competing objectives and constraints.
- We incorporate the Pareto optimization into the Monte Carlo tree search process and further design an anytime and non-myopic planner for in-situ decision-making.
- We provide in-depth algorithmic analysis which reveals bounding and converging behaviors of the search process.
- We perform thorough simulation evaluations with both synthetic and real-world data. Our results show that the robot exhibits desired composite behaviors which are

W. Chen and L. Liu are with the School of Informatics, Computing, and Engineering at Indiana University, Bloomington, IN 47408, USA. E-mail: {chenweiz, lantao}@iu.edu.

optimized from corresponding hybrid objectives.

II. RELATED WORK

Informative planning maximizes the collected information (informativeness) by exploring (partially) unknown environment during its sampling process [4, 17]. Comparing with the lawnmower based sweeping style sampling mechanism which focuses on spatial resolution, the informative planning method tends to achieve the spatial coverage quickly with the least estimation uncertainty [25]. Due to these reasons, the information planning has been widely used for the spatiotemporal environmental monitoring. To explore and learn the environment model, a commonly-used approach in spatial statistics is the Gaussian Process Regression [22]. Built on the learned environmental model, path and motion control can be carried out which is a critical ability for autonomous robots operating in unstructured environments [9, 16].

Representative informative planning approaches include, e.g., algorithms based on a recursive-greedy style [21, 25] where the informativeness is generalized as submodular function and a sequential-allocation mechanism is designed in order to obtain subsequent waypoints. This recursive-greedy framework has been extended later by incorporating obstacle avoidance and diminishing returns [4]. In addition, a differential entropy based framework [17] was proposed where a batch of waypoints can be obtained through dynamic programming. Recent work also reveals that online informative planning is possible [18]. The sampling data is thinned based on their contributions learned by a sparse variant of Gaussian Process. There are also methods optimizing over complex routing constraints (e.g., see [27, 31]).

Pareto optimization has been used in designing motion planners to optimize over the length of a path and the probability of collisions [6]. Recently, a sampling based method has also been proposed to generate Pareto-optimal trajectories for multi-objective motion planning [15]. In addition, multi-robot coordination also benefits from multi-objective optimization. The goals of different robots are simultaneously optimized [8, 14]. To balance the operation cost and the travel discomfort experienced by users, the multi-objective fleet routing algorithms compute the Pareto-optimal fleet operation plans [5].

Related work also includes the multi-objective reinforcement learning [24]. Particularly, the prior work multi-objective Monte Carlo tree search (MO-MCTS) is closely relevant to our method [30]. Unfortunately, MO-MCTS is computationally prohibitive and cannot be used for online planning framework. Vast computational resources are needed in order to maintain a global Pareto optimal set with all the best solutions obtained so far. In contrast, we develop a framework that maintains a local approximate Pareto optimal set in each node which can be processed in a much faster way. Our approach is also flexible and adaptive with regards to capturing environmental variabilities of different stages.

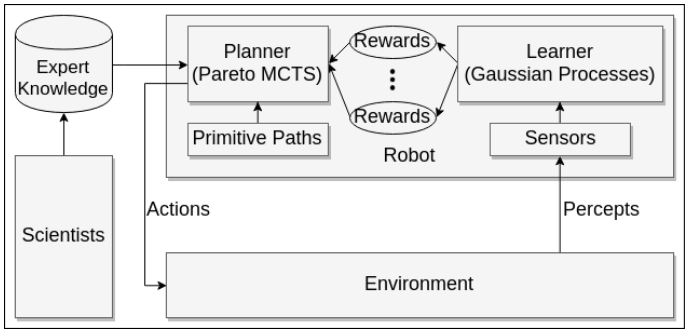


Fig. 2. System flow chart of Pareto MCTS for the online planning framework.

III. PRELIMINARIES

A. Pareto Optimality

Let $\boldsymbol{X}_k$ be a D-dimensional reward vector associated with a choice k and $X_{k,d}$ be the d-th element. The i-th choice is better than, or *dominating*, another choice j, denoted by $i \succ j$ or $j \prec i$, if and only if the following conditions are satisfied:

1) Any element of $\boldsymbol{X}_i$ is not smaller than the corresponding element in $\boldsymbol{X}_j$:
   $\forall d = 1, 2, \ldots, D, X_{i,d} \geqslant X_{j,d}$;
2) At lease one element of $\boldsymbol{X}_i$ is larger than the corresponding element in $\boldsymbol{X}_j$:
   $\exists d \in \{1, 2, \ldots, D\}$ such that $X_{i,d} > X_{j,d}$.

If only the first condition is satisfied, we say that choice i is *weakly-dominating* choice j, denoted by $i \succeq j$ or $j \preceq i$.

In some cases, neither $i \succeq j$ nor $j \succeq i$ hold. We say that choice i is *incomparable* with choice j ($i||j$) if and only if there exists one dimension d_1 such that $X_{i,d_1} > X_{j,d_1}$, and another dimension d_2 such that $X_{i,d_2} < X_{j,d_2}$. Also, we say that choice i is *non-dominated* by choice j ($i \not\prec j$ or $j \not\succ i$) if and only if there exists one dimension d such that $X_i^d > X_j^d$.

B. Multi-Objective Informative Planning

In the general case, *multi-objective informative planning* requires solving the following maximization problem:

$$\begin{aligned} \boldsymbol{a}^* = & \arg\max_{\boldsymbol{a} \in \mathcal{A}} \{I(\boldsymbol{a}), F_1(\boldsymbol{a}), \ldots, F_{D-1}(\boldsymbol{a})\}, \\ & \text{s.t. } C_{\boldsymbol{a}} \leq B, \end{aligned} \tag{1}$$

where $\boldsymbol{a}$ is a sequence of actions, $\mathcal{A}$ is the space of possible action sequences, B is a budget (e.g. time, energy, memory, or number of iterations), $C_{\boldsymbol{a}}$ is the cost of budget, and $I(\boldsymbol{a})$ is a function representing the information gathered by executing the action. $F_d(\boldsymbol{a}), d \in \{1, \ldots, D-1\}$ are other objective functions defining which types of behaviors are desired.

IV. APPROACH

A. Methodology Overview

An overview of our method is shown in Fig. 2. The planner computes an informative path using the Pareto MCTS algorithm based on the current estimation of the environment. As the robot travels along the path, it continuously receives observations of the target variables . Then, these newly acquired samples are

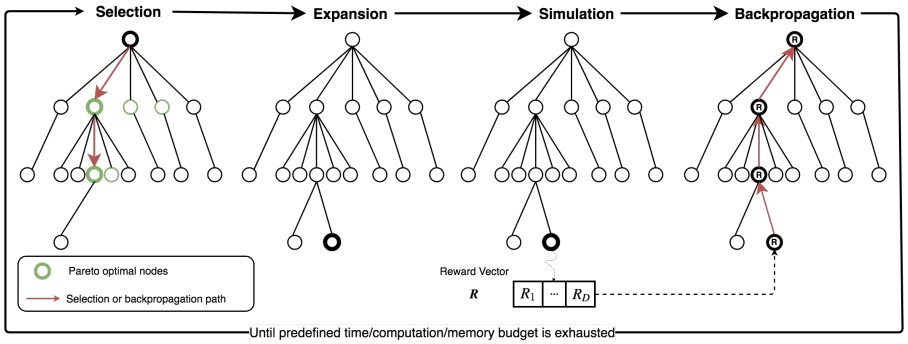


Fig. 3. Illustration of main steps of Pareto Monte Carlo tree search (Pareto MCTS).

used to refine its estimation of the environment, which in turn influences the planning in the next round.

The specific form of rewards depend on the applications. For example, in the informative planning problem, the rewards can be defined as the reduced amount of estimation entropy, accumulated amount of mutual information, etc. In hotspot exploration and exploitation task, the reward of a given path can be defined as the sum of rewards of all sampling points along the path. Pareto MCTS searches for the optimal actions for current state until a given time budget is exhausted. The inputs of Pareto MCTS are a set of available *primitive paths* at every possible state and the robot's current knowledge (estimation) of the world. It is worth noting that, the multi-objective framework also allows us to incorporate prior knowledge, as illustrated on the left side of Fig. 2.

B. Pareto Monte Carlo Tree Search

The proposed Pareto Monte Carlo tree search (Pareto MCTS) is a planning method for finding Pareto optimal decisions within a given horizon. The tree is built incrementally in order to expand the most promising subtrees first. This is done by taking advantage of the information gathered in previous steps. Each node in the search tree represents a state of the domain (in our context it is a location to be sampled), and the edges represent actions leading to subsequent states.

The framework of Pareto MCTS is outlined in Fig. 3. In each iteration, it can be broken down into four main steps:

1) *Selection*: starting at the root node, a child node selection policy (described later on) is applied recursively to descend through the tree until an expandable node with unvisited (unexpanded) children is encountered.
2) *Expansion*: an action is chosen from the aforementioned expandable node. A child node is constructed according to this action and connected to the expandable node.
3) *Simulation*: a simulation is run from the new node based on the predefined default policy. In our case, the default policy chooses a random action from all available actions. The reward vector of this simulation is then returned.
4) *Backpropagation*: the obtained reward is backed up or backpropagated to each visited node in the selection and expansion steps to update the attributes in those nodes. In our algorithm, each node stores the number of times it has been visited and the cumulative reward obtained by simulations starting from this node.

These steps are repeated until the maximum number of iterations is reached or a given time limit is exceeded.

The most challenging part of designing Pareto MCTS is the selection step. We want to select the most promising node to expand first in order to improve the searching efficiency. However, which node is the most promising is unknown so that the algorithm needs to estimate it. Therefore, in the selection step, one must balance exploitation of the recently discovered most promising child node and exploration of alternatives which may turn out to be a superior choice at later time.

In the case of scalar reward, the most promising node is simply the one with highest expected reward. However, the reward is herein a vector corresponding to multiple objectives. In the context of multi-objective optimization, the optima are no longer unique. Instead, there is a set of optimal choices, each of which is considered equally best. Below we define the Pareto optimal node set. The nodes in the Pareto optimal set are incomparable to each other and will not be dominated by any other nodes.

Definition 1 (Pareto Optimal Node Set)**.** *Given a set of nodes $\mathcal{V}$, a subset $\mathcal{P}^* \subset \mathcal{V}$ is the Pareto optimal node set, in terms of expected reward, if and only if*

$$\begin{cases} \forall v_i^* \in \mathcal{P}^* \text{ and } \forall v_j \in \mathcal{V}, v_i^* \nprec v_j, \\ \forall v_i^*, v_j^* \in \mathcal{P}^*, v_i^* || v_j^*. \end{cases}$$

We treat the node selection problem as a multi-objective multi-armed bandit problem. As shown in Alg. 1, the Pareto Upper Confidence Bound (Pareto UCB) for each child node is first computed according to Eq. (2). Then an approximate Pareto optimal set is built using the resulting Pareto UCB vectors (see the green nodes in Fig. 3). Finally, the best child node is chosen from the Pareto optimal set uniformly at random. Note that the reason for choosing the best child randomly is that the Pareto optimal solutions are considered equally optimal if no preference information is given. However, if domain knowledge is available or preference is specified, one can choose the most preferable child from the Pareto optimal set. For example, in the environmental monitoring task, one might expect the robot to explore the environment in the early stage to identify some important areas and spend more effort exploiting these areas in the later stage. In such case, we can

choose the Pareto optimal node with highest information gain in the beginning because the information gain based planning tends to cover the space quickly. Other types of rewards can be chosen later to concentrate on examining the details locally.

Note that this is different from weighting different objectives and solve a single-objective problem, because choosing a preferred solution from a given set of optimal solutions is often easier than determining the quantitative relationship among different objectives. For instance, it is difficult to quantify how much the information gain is more important than exploiting high-value areas in the hotspot monitoring task. However, given several choices, we may pick the most informative choice in the very beginning, and another one with highest target value related reward in a later stage.

Algorithm 1: Pareto MCTS

Function *Search*(s_0)
  create root node v_0 with state s_0
  while *within computational budget* **do**
    $v_{\text{expandable}} \leftarrow$ Selection(v_0)
    $v_{\text{new}} \leftarrow$ Expansion($v_{\text{expandable}}$)
    RewardVector $\leftarrow$ Simulation ($v_{\text{new}}.s$)
    Backpropagation(v_{new}, RewardVector)
  return MostVisitedChild(v_0)
Function *Selection*(v)
  while v *is fully expanded* **do**
    $v \leftarrow$ ParetoBestChild(v)
  return v
Function *ParetoBestChild*(v)
  compute Pareto UCB for each child k:

$$\boldsymbol{U}(k) = \frac{v_k.\boldsymbol{X}}{v_k.n} + \sqrt{\frac{4\ln n + \ln D}{2v_k.n}} \tag{2}$$

  build approximate Pareto optimal node set $v.\mathcal{P}$ based on $\boldsymbol{U}(k)$
  choose a child v_{best} from $v.\mathcal{P}$ uniformly at random
  return v_{best}

Note: We use v.attribute to represent a node attribute of v. In Eq. (2), $v_k.\boldsymbol{X}$ is the cumulative reward of the k-th child of node v, $v_k.n$ is the number of times that the k-th child has been visited, and D is the number of objectives. Some standard components, such as Simulation, Expansion, and Backpropagation, are not shown due to the lack of space.

C. Algorithm Analysis

As mentioned earlier, the node selection is the most critical step and almost determines the performance of the entire framework. Thus, here we spend effort analyzing this step to better understand its important properties. Although some (potentially) sub-optimal nodes may be selected inevitably, we show that the number of times choosing a sub-optimal node can be bounded logarithmically in Pareto MCTS. In addition, we want to know whether this anytime algorithm will converge to the optimal solution if enough time is given and whether a good solution can be returned if it is stopped midway. To answer this question, we show that the searching result of Pareto MCTS converges to the Pareto optimal choices at a polynomial rate.

Problem 1 (Node Selection)**.** *Consider a node v with K child nodes in a Pareto Monte Carlo search tree. At decision step n, a D-dimensional random reward $\boldsymbol{X}_{k,n_k}$ will be returned after selecting child v_k. Successive selections of child v_k yield rewards $\boldsymbol{X}_{k,1}, \boldsymbol{X}_{k,2}, \ldots$, which are drawn from an unknown distribution with unknown expected reward $\boldsymbol{\mu}_k$. A policy is an algorithm that chooses a child node based on the sequence of past selections and obtained rewards. The goal of a policy is to minimize the number of times choosing a sub-optimal node.*

In Pareto MCTS, node selection only happens after all child nodes have been expanded. In other words, there is an initialization step in which each node has been selected once. For easy reference, we summarize the node selection policy below.

Policy 1. *Given a node selection problem as Problem 1, choose each child node once in the first K steps. After that, build an approximate Pareto optimal node set based on the following upper confidence bound vector:*

$$\boldsymbol{U}(k) = \bar{\boldsymbol{X}}_{k,n_k} + \sqrt{\frac{4\ln n + \ln D}{2n_k}}, \tag{3}$$

where K is the number of child nodes, n_k is the number of times child k has been selected so far, $\bar{\boldsymbol{X}}_{k,n_k}$ is the average reward obtained from child k, D is the number of dimensions of the reward, and $n = \sum_{k=1}^{K} n_k$.

In the following proof, we shall use the concept of most dominant optimal node originated from the ϵ-dominance concept [12] of multi-objective optimization. Intuitively, the most dominant optimal node of a given node is the one in the (estimated) Pareto optimal set which is the "farthest away" from the given node.

Definition 2 (Most Dominant Optimal Node)**.** *Given a node v_k and a node set $\mathcal{V}$ such that $\forall v_{k'} \in \mathcal{V}, v_{k'} \succ v_k$. For all $v_{k'} \in \mathcal{V}$, there exists exactly one minimum positive constant $\epsilon_{k'}$ such that*

$$\epsilon_{k'} = \min\{\epsilon | \exists d \in \{1, 2, \ldots, D\} \text{ s.t. } \mu'_{k,d} + \epsilon > \mu_{k,d}\}.$$

Let the index of the maximum $\epsilon_{k'}$ be k^*,

$$k^* = \arg\max_{k'} \epsilon_{k'},$$

then the most dominant optimal node is v_{k^*}.

Throughout the paper, symbols related to the most dominant optimal node will be indexed by a star(*). As in [11], we allow the expected average rewards to drift as a function of time and our main assumption is that it will converge pointwise. Here we introduce two assumptions so that the later proof can exploit.

Assumption 1 (Convergence of Expected Average Rewards)**.** *The expectations of the average rewards $\mathbb{E}[\bar{\boldsymbol{X}}_{k,n_k}]$ converge pointwise to the limit $\boldsymbol{\mu}_k$ for all child nodes:*

$$\boldsymbol{\mu}_k = \lim_{n_k \to \infty} \mathbb{E}[\bar{\boldsymbol{X}}_{k,n_k}]. \tag{4}$$

For a sub-optimal node v_k and its most dominant optimal node v_{k^*} from $\mathcal{P}^*$, we define $\boldsymbol{\Delta}_k = \boldsymbol{\mu}^* - \boldsymbol{\mu}_k$ to denote their difference.

Assumption 2. *Fix $1 \le k \le K$ and $1 \le d \le D$. Let $\{\mathcal{F}_{k,t,d}\}_t$ be a filtration such that $\{X_{k,t,d}\}_t$ is $\{\mathcal{F}_{k,t,d}\}$-adapted and $X_{k,t,d}$ is conditionally independent of $\mathcal{F}_{k,t+1,d}, \mathcal{F}_{k,t+2,d}, \dots$ given $\mathcal{F}_{k,t-1,d}$.*

For the sake of simplifying the notation, we define $\boldsymbol{\mu}_{k,n_k} = \mathbb{E}[\bar{\boldsymbol{X}}_{k,n_k}]$ and $\boldsymbol{\delta}_{k,n_k} = \boldsymbol{\mu}_{k,n_k} - \boldsymbol{\mu}_k$ as the residual for the drift. Clearly, $\lim_{n_k \to \infty} \boldsymbol{\delta}_{k,n_k} = \mathbf{0}$. By definition, $\forall \xi > 0$, $\exists N_0(\xi)$ such that $\forall n_k \ge N_0(\xi)$, $\forall d \in \{1, 2, \dots, D\}$, $|\delta_{k,n_k,d}| \le \xi \Delta_{k,d}/2$. We present the first theorem below.

Theorem 1. *Consider Policy 1 applied to the node selection Problem 1. Suppose Assumption 1 is satisfied. Let $T_k(n)$ denote the number of times child node v_k has been selected in the first n steps. If child node v_k is a sub-optimal node (i.e. $v_k \notin \mathcal{P}^*$), then $\mathbb{E}[T_k(n)]$ is logarithmically bounded:*

$$\mathbb{E}[T_k(n)] \le \frac{8 \ln n + 2 \ln D}{(1-\xi)^2 (\min_{k,d} \Delta_{k,d})^2} + N_0(\xi) + 1 + \frac{\pi^2}{3}. \tag{5}$$

Proof: Proof is provided in Appendix A. ∎

The following lemma gives a lower bound on the number of times each child node being selected.

Lemma 1. *There exists positive constant ρ such that $\forall k, n, T_k(n) \ge \lceil \rho \log(n) \rceil$.*

The upcoming lemma states that the average reward will concentrate around its expectation after enough node selection steps.

Lemma 2 (Tail Inequality)**.** *Fix arbitrary $\eta > 0$ and let $\sigma = 9\sqrt{\frac{2\ln(2/\eta)}{n}}$. There exists $N_1(\eta)$ such that $\forall n \ge N_1(\eta), \forall d \in \{1, \dots, D\}$, the following bounds hold true:*

$$\mathbb{P}(\bar{X}_{n,d} \ge \mathbb{E}[\bar{X}_{n,d}] + \sigma) \le \eta, \tag{6}$$

$$\mathbb{P}(\bar{X}_{n,d} \le \mathbb{E}[\bar{X}_{n,d}] - \sigma) \le \eta. \tag{7}$$

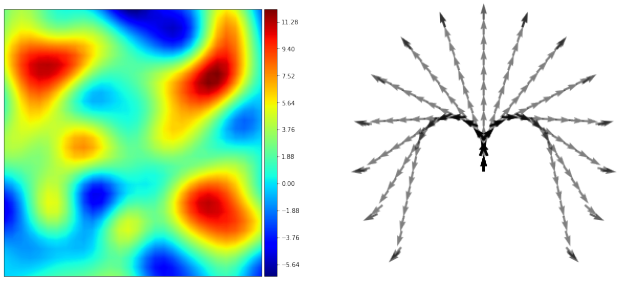
Correctness of Lemma 1 and Lemma 2 is provided in [11].

Theorem 2 (Convergence of Failure Probability)**.** *Consider the node selection policy described in Algorithm 1 applied to the root node. Let I_t be the selected child node and $\mathcal{P}^*$ be the Pareto optimal node set. Then,*

$$\mathbb{P}(I_t \notin \mathcal{P}^*) \le C t^{-\frac{\rho}{2}\left(\frac{\min_{k,d} \Delta_{k,d}}{36}\right)^2}, \tag{8}$$

with some constant C. In particular, it holds that $\lim_{t \to \infty} \mathbb{P}(I_t \notin \mathcal{P}^*) = 0$

Proof: Proof is provided in Appendix B. ∎



(a) (b)

Fig. 4. (a) Environment with hotspots used as ground truth. The heat map represents level of interest. (b) Primitive paths for a robot.

Theorem 2 shows that, at the root node, the probability of choosing a child node (and corresponding action) which is not in the Pareto optimal set converges to zero at a polynomial rate as the number of node selection grows.

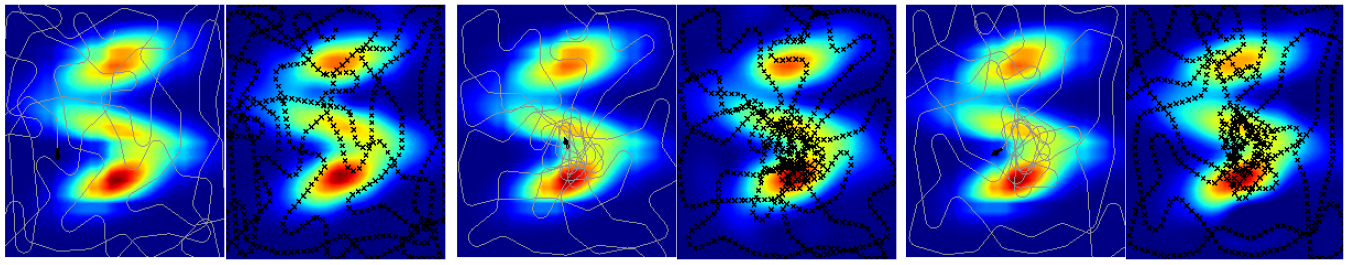
V. EXPERIMENTS

To thoroughly evaluate our framework, we have compared different methods in extensive simulations using both synthetic data and real-world data, where the basic scenario is the hotspot monitoring task via informative planning for a robot. The ideal behavior for the robot is to first explore the environment to discover the hotspots and then exploit these important areas to collect more valuable samples. This is a bi-objective task although our algorithm is suitable for multi-objective tasks in general. We choose this scenario for comparison (and illustration) purpose, since one of the comparing methods can only handle bi-objective case. In addition, hotspot monitoring task can be easily visualized for interpretation.

We have compared our algorithm with two other baseline methods. The first method is the Monte Carlo tree search with information-theoretic objective, which has been successfully applied to planetary exploration mission [1], environment exploration [2, 7], and monitoring spatiotemporal process [19]. We called the method *information MCTS*. The second method is a upper confidence bound based online planner [28] which balances exploration and exploitation in a near-optimal manner with appealing no-regret properties. However, when choosing an optimal trajectory, only the primitive paths of current state are considered in their model. For comparison, we modify and extend this model to MCTS by using the upper confidence bound (see section IV of [28]) as the reward function of MCTS, which is called *UCB MCTS*.

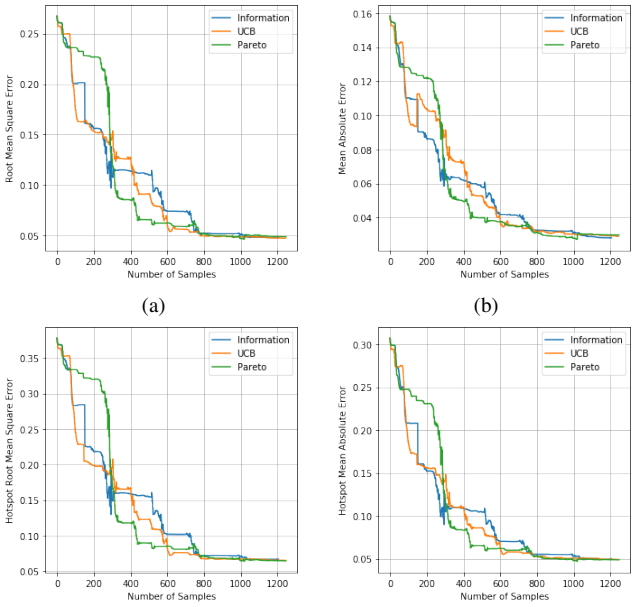
Rewards: As for the reward function, we choose variance reduction as the information-theoretic objective as in [3, 10]. The other reward is devised as adding up the predicted values of samples along the path, which encourages the robot to visit high-value areas more frequently.

Metrics: In the informative planning context, our goal is to minimize the root mean square error (RMSE) between the estimated environmental state (using GP prediction) and the ground truth. However, in our hotspot monitoring task, we are more concerned with the modeling errors in high-value areas than the entire environment. Therefore, a *hotspot RMSE* is



(a) (b) (c) (d) (e) (f)

Fig. 5. The arrow represents the robot. (a)(c)(e) The line shows the resulting path of each algorithm with the underlying environment as the background. (b)(d)(f) Robot's estimation of the target value and collected samples. Red represents high value and blue indicates low value. (a)(b) Information MCTS. (c)(d) UCB MCTS. (e)(f) Pareto MCTS.



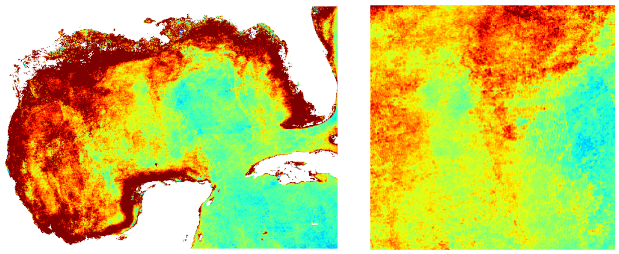
(a) (b) (c) (d)

Fig. 6. The resulting error of the three algorithm in the synthetic problem. (a) Root mean square error. (b) Mean absolute error. (c) Hotspot root mean square error. (d) Hotspot mean absolute error.

employed for better evaluating the "exploitation" performance. We classify the areas with target values higher than the median as hotspots and calculate the RMSE within these hotspots. In addition, larger errors in unimportant areas are acceptable in this task and RMSE tends to penalize larger errors in a uniform way. Therefore, we also evaluate the methods using mean absolute error (MAE). Similarly, we introduced hotspot MAE to highlight algorithms' performance in the important areas. Last but not least, the *percentage of samples* in hotspots measures the quality of the collected data. Ideally, the robot should locate the important areas as soon as possible and gather more valuable samples in these areas. Also, if there are multiple hotpots, the robot should visit as many hotspots as possible instead of getting stuck in one specific area.

A. Synthetic Problems

The robot is tasked to monitor several hotspots in an unknown 10 km $\times$ 10 km environment, illustrated in Fig. 4a. The hotspots to be monitored are specified by three Gaussian sources with random placement and parameters. At each position, the robot has 15 Dubins paths [13] as its primitive paths. Fig. 4b illustrates an example of available primitive paths.

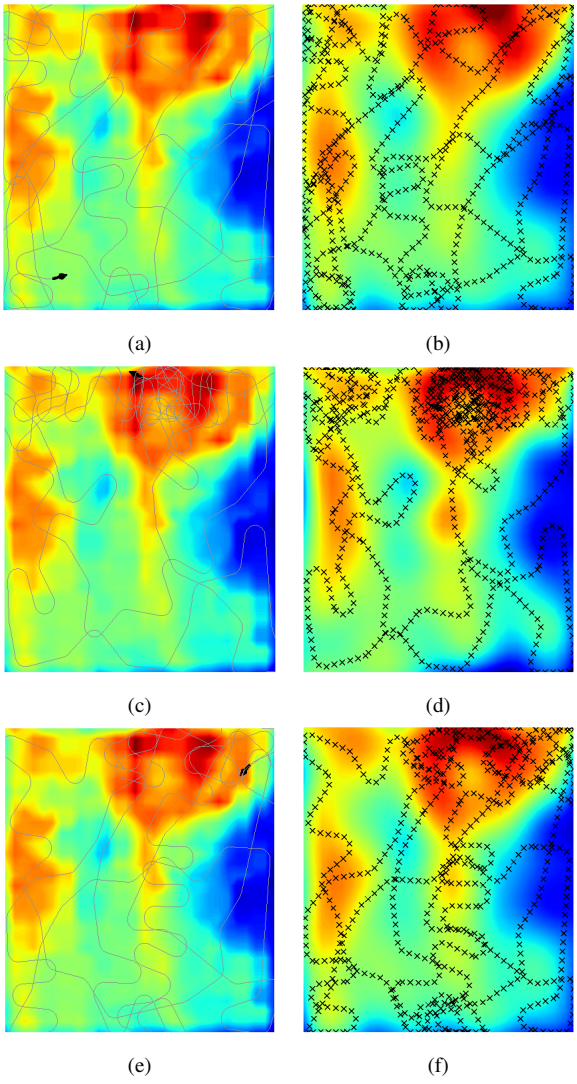


(a) (b)

Fig. 7. (a) CDOM raw data of Gulf of Mexico. (b) A cropped region near Louisiana Wetlands.

Fig. 5 shows the the ground truth and prediction with the robot's path and collected samples. As expected, the information MCTS tends to cover the whole space and collect the samples uniformly, because less information will be gained from a point once it has been observed. On the contrary, UCB MCTS spent a small amount of effort exploring the environment during the initial phase in order to reduce the uncertainty of the hotspot estimation. After that, it tends to greedily wander around the high-value areas. This phenomenon is consistent with the behavior of the UCB algorithm in the multi-armed bandit problem in which the best machine will be played much more times than sub-optimal machines. We noticed that the bias term in UCB-replanning [28] also increases with the mission time, which means that, given enough time, the robot will still try to explore other areas. However, in our experiments, the task duration could not be set too long due to the scalability of the GP. Pareto MCTS simultaneously optimize all the objectives. As a result, more samples can be found in the upper hotspot (see Fig. 5f and Fig. 5d).

Fig. 6 presents the global error and hotspot error. The difference between the global error and the hotspot error is negligible because the most important variability is concentrated in the hotspot areas. We notice that, in the first 200 samples, the performance of Pareto MCTS is inferior to other two methods. In fact, this is because Pareto MCTS has not yet discovered any particular hotspot in the initial exploration phase. Once it finds the a hotspot and starts exploiting that hotspot, the error curve



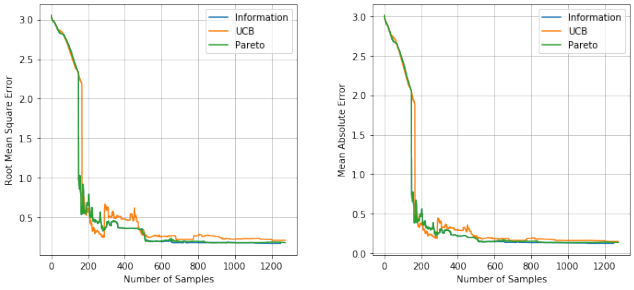
(a) (b) (c) (d) (e) (f)

Fig. 8. The resulting path, collected samples, and prediction of each algorithm in the chromophoric dissolved organic material monitoring problem. Blue and red correspond to low and high value, respectively. The arrow represents the robot and the crosses are the sampled data. (a) Path and the ground truth of information MCTS. (b) Collected samples and the estimated hotspot map from information MCTS. (c) Path and the ground truth of UCB MCTS. (d) Collected samples and estimation of UCB MCTS. (e) Path and the ground truth of Pareto MCTS. (f) Collected samples and estimation of Pareto MCTS.

drops drastically, surpassing the other two methods at about 300 samples. This property is consistent with our motivation. It is also obvious that our method in general has the steepest error reduction rate, which is particularly important in monitoring highly dynamic environment.

B. Chromophoric Dissolved Organic Material Monitoring

We now demonstrate our proposed approach using the chromophoric dissolved organic material (CDOM) data at the Gulf of Mexico, provided by National Oceanic and Atmospheric Administration (NOAA). The concentration of CDOM has a significant effect on biological activity in aquatic systems. Very high concentrations of CDOM can affect photosynthesis and inhibit the growth of phytoplanktons. Fig. 7a is the raw data



(a) (b)

Fig. 9. The root mean square error and mean absolute error of the three algorithms in the chromophoric dissolved organic material monitoring problem. (a) Root mean square error. (b) Mean absolute error. The blue line is visually overlapped with the green line. They are separated after zooming in, but the difference is negligible.

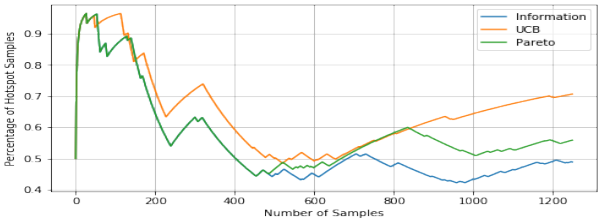


Fig. 10. Percentage of hotspot samples of the three algorithms in the chromophoric dissolved organic material monitoring problem.

of CDOM. We have cropped a smaller region with a higher variability of the target value (Fig. 7b). Due to the scalability issue of the GP, the raw data (300×300 grids) is down-sampled to (30×30).

Fig. 8 reveals similar sample distribution patterns as in the synthetic data. Specifically, the information MCTS features good spatial coverage. UCB MCTS prefers to stay at the hotspot with highest target value after a rough exploration of the environment. Pareto MCTS tends to compromise between hotspot searching and hotspot close examination. As a result, it exhibits interesting winding paths in some important areas (see Fig. 8f). This allows the robot to collect more samples in those areas without losing too much information gain.

In this experiment, we also show how to incorporate prior knowledge in the Pareto MCTS. In robotic environmental monitoring, the robot needs to explore the environment extensively in the early stage. To this end, we always choose the most informative action from the Pareto optimal set at the beginning (first 400 samples), which makes Pareto MCTS degenerate to information MCTS. As shown in Fig. 9, the root mean square error of information MCTS (blue line) and that of Pareto MCTS (green line) visually overlap. This implies that there is almost no loss in global modeling error. At the same time, the percentage of samples collected from hotspots has increased. Fig. 10 shows the percentage of hotspot samples. These results validate the benefits of multi-objective informative planning and Pareto MCTS.

VI. CONCLUSION

This paper presents a Pareto multi-objective optimization based informative planning approach. We show that the searching result of Pareto MCTS converges to the Pareto optimal actions at a polynomial rate, and the number of times choosing a sub-optimal node in the course of tree search has a logarithmic bound. Our method allows the robot to adapt to the target environment and adjust its concentrations on environmental exploration versus exploitation based on its knowledge (estimation) of the environment. We validate our approach in a hotspot monitoring task using real-world data and the results reveal that our algorithm enables the robot to explore the environment and, at the same time, visit hotspots of high interests more frequently.

APPENDIX

A. Proof to Theorem 1

Let variable I_t be the index of the selected child node at decision step t and $\mathbf{1}\{\cdot\}$ be a Boolean predicate function. Let the bias term in Eq. (3) be $c_{t,s} = \sqrt{\frac{4\ln t + \ln D}{2s}}$. Then $\forall l > 0$ and $l \in \mathbb{Z}^+$, for any sub-optimal node v_k, we have the upper bound $T_k(n)$.

$$\begin{aligned}
T_k(n) =& l + \sum_{t=K+1}^{n} \mathbf{1}\{I_t = k\} \\
\leq& l + \sum_{t=K+1}^{n} \mathbf{1}\{I_t = k, T_k(t-1) \geq l\} \\
\leq& l + \sum_{t=K+1}^{n} \mathbf{1}\{\bar{\boldsymbol{X}}^*_{T_j(t-1)} + c_{t-1,T^*(t-1)} \\
& \nsucc \bar{\boldsymbol{X}}_{k,T_k(t-1)} + c_{t-1,T_k(t-1)}, T_k(t-1) \geq l\} \\
\leq& l + \sum_{t=1}^{\infty}\sum_{s=1}^{t-1}\sum_{s_k=l}^{t-1} \mathbf{1}\{\bar{\boldsymbol{X}}^*_s + c_{t,s} \nsucc \bar{\boldsymbol{X}}_{k,s_k} + c_{t,s_k}\}
\end{aligned}$$

$\bar{\boldsymbol{X}}^*_s + c_{t,s} \nsucc \bar{\boldsymbol{X}}_{k,s_k} + c_{t,s_k}$ implies that at least one of the following must hold:

$$\bar{\boldsymbol{X}}^*_s \nsucc \boldsymbol{\mu}^*_s - c_{t,s} \tag{9}$$

$$\bar{\boldsymbol{X}}_{k,s_k} \nprec \boldsymbol{\mu}_{k,s_k} - c_{t,s} \tag{10}$$

$$\boldsymbol{\mu}^*_s \nsucc \boldsymbol{\mu}_{k,s} + 2c_{t,s_k} \tag{11}$$

Otherwise, If Eq. (9), (10) are false, then Eq. (11) is true.

We bound the probability of events Eq. (9) (10) using Chernoff-Hoeffding Bound and Union Bound.

$$\begin{aligned}
&\mathbb{P}(\bar{\boldsymbol{X}}^*_s \nsucc \boldsymbol{\mu}^*_s - c_{t,s}) \\
=&\mathbb{P}\left((\bar{X}^*_{s,1} < \mu^*_{s,1} - c_{t,s}) \vee \cdots \vee (\bar{X}^*_{s,D} < \mu^*_{s,D} - c_{t,s})\right) \\
\leq& \sum_{d=1}^{D} \mathbb{P}\left(\bar{X}^*_{s,d} < \mu^*_{s,d} - c_{t,s}\right) && \text{(Union Bound)} \\
\leq& \sum_{d=1}^{D} \frac{1}{D} t^{-4} = t^{-4} && \text{(Chernoff-Hoeffding Bound)}
\end{aligned}$$

Similarly, $\mathbb{P}(\bar{\boldsymbol{X}}_{i,s_k} \nprec \boldsymbol{\mu}_{i,s_k} - c_{t,s}) \leq t^{-4}$

Let

$$l = \max\left\{\left\lceil \frac{8\ln t + 2\ln D}{(1-\xi)^2 \min_{k,d} \Delta^2_{k,d}} \right\rceil, N_0(\xi)\right\}.$$

Since $s_{k,d} \geq l_0$, (11) is false.

Therefore, plugging the above results into the bound on $T_k(n)$ and taking expectations of both sides, we get

$$\begin{aligned}
\mathbb{E}[T_k(n)] \leq& \left\lceil \frac{8\ln t + 2\ln D}{(1-\xi)^2 \min_{k,d} \Delta^2_{k,d}} \right\rceil + N_0(\xi) + \sum_{t=1}^{\infty}\sum_{s=1}^{t-1}\sum_{s_k=l}^{t-1} \\
& \left(\mathbb{P}(\bar{\boldsymbol{X}}^*_s \nsucc \boldsymbol{\mu}^*_s - c_{t,s}) + \mathbb{P}(\bar{\boldsymbol{X}}_{k,s_k} \nprec \boldsymbol{\mu}_{k,s_k} - c_{t,s})\right) \\
\leq& \frac{8\ln t + 2\ln D}{(1-\xi)^2 \min_{d} \Delta^2_{k,d}} + N_0(\xi) + 1 + \frac{\pi^2}{3},
\end{aligned}$$

which concludes the proof.

B. Proof to Theorem 2

Let k be the index of a sub-optimal node. Then $\mathbb{P}(I_t \notin \mathcal{P}^*) \leq \sum_{v_k \notin \mathcal{P}^*} \mathbb{P}\left(\bar{\boldsymbol{X}}_{k,T_k(t)} \nprec \bar{\boldsymbol{X}}^*_{T^*(t)}\right)$. Note that $\bar{\boldsymbol{X}}_{k,T_k(t)} \nprec \bar{\boldsymbol{X}}^*_{T^*(t)}$ implies

$$\bar{\boldsymbol{X}}_{k,T_k(t)} \nprec \boldsymbol{\mu}_k + \frac{\boldsymbol{\Delta}_k}{2}, \tag{12}$$

or

$$\bar{\boldsymbol{X}}^*_{T^*(t)} \nsucc \boldsymbol{\mu}^* + \frac{\boldsymbol{\Delta}_k}{2}. \tag{13}$$

Otherwise, suppose Eq. (12) and (13) do not hold, we have $\bar{\boldsymbol{X}}_{k,T_k(t)} \prec \bar{\boldsymbol{X}}^*_{T^*(t)}$ which yields a contradiction. Hence,

$$\begin{aligned}
&\mathbb{P}\left(\bar{\boldsymbol{X}}_{k,T_k(t)} \nprec \bar{\boldsymbol{X}}^*_{T^*(t)}\right) \\
\leq& \underbrace{\mathbb{P}(\bar{\boldsymbol{X}}_{k,T_k(t)} \nprec \boldsymbol{\mu}_k + \frac{\boldsymbol{\Delta}_k}{2})}_{\text{first term}} + \underbrace{\mathbb{P}(\bar{\boldsymbol{X}}^*_{T^*(t)} \nsucc \boldsymbol{\mu}^* + \frac{\boldsymbol{\Delta}_k}{2})}_{\text{second term}}.
\end{aligned}$$

Here we show how to bound the first term:

$$\begin{aligned}
&\mathbb{P}(\bar{\boldsymbol{X}}_{k,T_k(t)} \nprec \boldsymbol{\mu}_k + \frac{\boldsymbol{\Delta}_k}{2}) \\
\leq& \sum_{d=1}^{D} \mathbb{P}(\bar{X}_{k,T_k(t),d} > \mu_{k,d} + \frac{\Delta_{k,d}}{2}) && \text{(Union Bound)} \\
\leq& \sum_{d=1}^{D} \mathbb{P}(\bar{X}_{k,T_k(t),d} \geq \mu_{k,T_k(t),d} - \underbrace{|\delta_{k,T_k(t),d}|}_{\text{converges to 0}} + \frac{\Delta_{k,d}}{2}) \\
\leq& \sum_{d=1}^{D} \mathbb{P}(\bar{X}_{k,T_k(t),d} \geq \mu_{k,T_k(t),d} + \frac{\Delta_{k,d}}{4}) \\
\leq& \sum_{d=1}^{D} \text{constant}(\frac{1}{t})^{\frac{\rho}{2}\left(\frac{\min_{k,d}\Delta_{k,d}}{36}\right)^2} && \text{(Lemma 2)}
\end{aligned}$$

The last step makes use of Lemma 1 and Lemma 2.

The second term can be bounded in a similar way. Finally, an integration of the bounds shows that the failure probability converges to 0 at a polynomial rate as the number of selection goes to infinity.

References

[1] Akash Arora, Robert Fitch, and Salah Sukkarieh. An approach to autonomous science by modeling geological knowledge in a bayesian framework. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3803–3810. IEEE, 2017.

[2] Graeme Best, Oliver M Cliff, Timothy Patten, Ramgopal R Mettu, and Robert Fitch. Dec-mcts: Decentralized planning for multi-robot active perception. *The International Journal of Robotics Research*, 38(2-3):316–337, 2019.

[3] Jonathan Binney and Gaurav S Sukhatme. Branch and bound for informative path planning. In *IEEE International Conference on Robotics and Automation*, pages 2147–2154. IEEE, 2012.

[4] Jonathan Binney, Andreas Krause, and Gaurav S. Sukhatme. Optimizing waypoints for monitoring spatiotemporal phenomena. *International Journal on Robotics Research (IJRR)*, 32(8):873–888, 2013.

[5] Michal Cáp and Javier Alonso-Mora. Multi-objective analysis of ridesharing in automated mobility-on-demand. In *Robotics: Science and Systems*, 2018.

[6] Shushman Choudhury, Christopher M Dellin, and Siddhartha S Srinivasa. Pareto-optimal search over configuration space beliefs for anytime motion planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3742–3749. IEEE, 2016.

[7] Micah Corah and Nathan Michael. Efficient online multi-robot exploration via distributed sequential greedy assignment. In *Proceedings of robotics: science and systems*, 2017.

[8] Robert Ghrist, Jason M OKane, and Steven M LaValle. Pareto optimal coordination on roadmaps. In *Algorithmic foundations of robotics VI*, pages 171–186. Springer, 2004.

[9] Carlos Ernesto Guestrin. *Planning Under Uncertainty in Complex Structured Environments*. PhD thesis, Stanford, CA, USA, 2003. AAI3104233.

[10] Geoffrey A Hollinger and Gaurav S Sukhatme. Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33(9):1271–1287, 2014.

[11] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.

[12] Joshua B Kollat, Patrick M Reed, and Joseph R Kasprzyk. A new epsilon-dominance hierarchical bayesian optimization algorithm for large multiobjective monitoring network design problems. *Advances in Water Resources*, 31(5): 828–845, 2008.

[13] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[14] Steven M LaValle and Seth A Hutchinson. Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation*, 14(6):912–925, 1998.

[15] Jeongseok Lee, Daqing Yi, and Siddhartha S Srinivasa. Sampling of pareto-optimal trajectories using progressive objective evaluation in multi-objective motion planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE, 2018.

[16] Naomi E Leonard, Derek A Paley, Russ E Davis, David M Fratantoni, Francois Lekien, and Fumin Zhang. Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in monterey bay. *Journal of Field Robotics*, 27(6):718–740, 2010.

[17] Kian Hsiang Low. *Multi-robot Adaptive Exploration and Mapping for Environmental Sensing Applications*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2009.

[18] Kai-Chieh Ma, Lantao Liu, Hordur K Heidarsson, and Gaurav S Sukhatme. Data-driven learning and planning for environmental sampling. *Journal of Field Robotics*, 35(5):643–661, 2018.

[19] Roman Marchant, Fabio Ramos, Scott Sanner, et al. Sequential bayesian optimisation for spatial-temporal monitoring. In *UAI*, pages 553–562, 2014.

[20] Seth McCammon and Geoffrey A Hollinger. Topological hotspot identification for informative path planning with a marine robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

[21] Alexandra Meliou, Andreas Krause, Carlos Guestrin, and Joseph M. Hellerstein. Nonmyopic informative path planning in spatio-temporal models. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, pages 602–607, 2007.

[22] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[23] John H Reif. Complexity of the mover's problem and generalizations. In *20th Annual Symposium on Foundations of Computer Science*, pages 421–427, 1979.

[24] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

[25] Amarjeet Singh, Andreas Krause, Carlos Guestrin, William Kaiser, and Maxim Batalin. Efficient planning of informative paths for multiple robots. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 2204–2211, 2007.

[26] Amarjeet Singh, Andreas Krause, Carlos Guestrin, and William J Kaiser. Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research*, 34:707–755, 2009.

[27] Daniel E Soltero, Mac Schwager, and Daniela Rus. Generating informative paths for persistent sensing in unknown environments. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2172–2179. IEEE, 2012.

[28] Wen Sun, Niteesh Sood, Debadeepta Dey, Gireeja Ranade, Siddharth Prakash, and Ashish Kapoor. No-regret replanning under uncertainty. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6420–6427. IEEE, 2017.

[29] Sara Susca, Francesco Bullo, and Sonia Martinez. Monitoring environmental boundaries with a robotic sensor network. *IEEE Transactions on Control Systems Technology*, 16(2):288–296, 2008.

[30] Weijia Wang and Michele Sebag. Multi-objective monte-carlo tree search. In *Asian conference on machine learning*, volume 25, pages 507–522, 2012.

[31] Jingjin Yu, Mac Schwager, and Daniela Rus. Correlated orienteering problem and its application to informative path planning for persistent monitoring tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 342–349. IEEE, 2014.