Measurement Maximizing Adaptive Sampling with Risk Bounding Functions

Benjamin Ayton, Brian Williams
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{aytonb, williams}@mit.edu

Richard Camilli
Woods Hole Oceanographic Institute
rcamilli@whoi.edu

Abstract

In autonomous exploration a mobile agent must adapt to new measurements to seek high reward, but disturbances cause a probability of collision that must be traded off against expected reward. This paper considers an autonomous agent tasked with maximizing measurements from a Gaussian Process while subject to unbounded disturbances. We seek an adaptive policy in which the maximum allowed probability of failure is constrained as a function of the expected reward. The policy is found using an extension to Monte Carlo Tree Search (MCTS) which bounds probability of failure. We apply MCTS to a sequence of approximating problems, which allows constraint satisfying actions to be found in an anytime manner. Our innovation lies in defining the approximating problems and replanning strategy such that the probability of failure constraint is guaranteed to be satisfied over the true policy. The approach does not need to plan for all measurements explicitly or constrain planning based only on the measurements that were observed. To the best of our knowledge, our approach is the first to enforce probability of failure constraints in adaptive sampling. Through experiments on real bathymetric data and simulated measurements, we show our algorithm allows an agent to take dangerous actions only when the reward justifies the risk. We then verify through Monte Carlo simulations that failure bounds are satisfied.

Introduction

A common mission in environment exploration is identification and confirmation of high reward regions. In underwater exploration, for example, autonomous vehicles may be tasked with locating regions with high temperatures, algal and plankton blooms, or high concentrations of pollutants or hydrocarbons for the purpose of identifying suitable locations for follow-up studies. When these follow-up studies are expensive and time consuming, for example involving the transportation and setup of expensive equipment, it is insufficient to minimize global uncertainty because the costs associated with performing the follow-up study at a location falsely believed to be valuable may exceed the costs of the initial autonomous sampling mission. Instead, samples must be taken at potential sites to confirm their importance, with a higher measurement being more valuable.

Such missions are performed because the environment is not well understood. New measurements contribute to an improved understanding as they are received, and impact the future of the mission. We require an adaptive system that is capable of updating its plan in response to new measurements to direct it towards high reward locations.

However, disturbances and noises acting on an agent make safe autonomous exploration difficult. In underwater applications, the seafloor surface is often known relatively well, but uncertainty in position accumulates due to unknown currents and inaccuracy in on-board inertial navigation. Navigating close to obstacles incurs some probability of collision, leading to failure, but it is often impossible to conduct a mission without accepting some level of risk. There is therefore a tradeoff between allowed risk and expected reward over which the autonomous system should be able to reason.

In this paper, we describe a method of planning an adaptive policy that maximizes the expected reward of samples, while limiting the probability of failure of the policy. We define a method of specifying tolerance for failure as a function of reward through a *risk bounding function*, and enforce the *chance constraint* that the expected rate of failure is bounded by the risk bounding function applied to the expected reward. Our method is based on Monte Carlo Tree Search (MCTS), which allows a solution to be found in an anytime manner, making it suitable for on-board autonomy or missions with tight time constraints. Furthermore, we are able to provably enforce the chance constraint without planning for all measurements explicitly, and without limiting allowed probability of failure based only on observations.

Related Work

Adaptive sampling tasks an agent with exploring an environment that is unknown. The environment is either characterized by a known uncertainty field (Hollinger and Sukhatme 2014; Yilmaz et al. 2008), or described by a stochastic process such as a Gaussian Process (GP) (Binney, Krause, and Sukhatme 2010; Krause, Singh, and Guestrin 2008; Low, Dolan, and Khosla 2009). It is typical to discretize available actions and perform discrete space search, though notable exceptions exist, including using Rapidly Exploring Random Trees (Hollinger and Sukhatme 2014) or genetic algorithms (Hitz et al. 2017). We will assume a GP model and

a discrete space of actions.

When tasked with maximizing information measures in a GP, the information depends only on the locations of samples and not their values (Binney, Krause, and Sukhatme 2010; Krause and Guestrin 2007). It follows that replanning in response to new information is not necessary, and sampling may be directed to regions where the GP is uncertain, but nonetheless believed to be low. In contrast, we maximize measurements of the GP, which depends explicitly on observations and requires an adaptive policy.

GP level set estimation is closer to our objective in that it selects sample locations to determine where a GP lies above or below a threshold (Bryan et al. 2006; Gotovos et al. 2013). However, no preference is given towards identifying regions above the threshold, nor is there consideration of the magnitude of the GP in high value regions. Our goal to detect locations with high values is similar to that of active search (Garnett et al. 2012), though we consider reward to be a real number instead of binary.

Even with discrete action and state spaces, the problem quickly becomes intractable as the search tree branches in both actions and the full history of observations. Fixed horizon planning strategies that consider only the next few actions (Krause, Singh, and Guestrin 2008; Marchant et al. 2014; Singh, Nowak, and Ramanathan 2006) cannot guarantee chance constraints, as no actions may be possible that satisfy failure bounds late in the policy, and setting a probability threshold for each action can lead to highly suboptimal policies. An entire policy that branches on measurements is found in the work by Low, Dolan, and Khosla (2009), though their experimental results imply small action spaces and relatively few measurements, whereas we consider on the order of 20 actions.

An alternative approach used by Hitz et al. (2017) is to plan and begin execution of a full policy that does not depend on measurement outcomes but satisfies a cost constraint. The policy is updated by replanning in response to new measurements. We adopt a similar approach, while the key differences between this work and Hitz et al. in this regard are that our chance constraint applies to the policy as a whole and depends on measurement outcomes. To the best of our knowledge, this is the first time chance constraints, or generally constraints on an expectation over a policy, have been applied to adaptive sampling.

Chance constrained planning has been examined for motion planning (Blackmore et al. 2010; Ono and Williams 2008), but bounds are applied to non-adaptive plans. While chance constraint satisfaction is guaranteed if it is satisfied for every outcome, the result can be highly suboptimal, and may not even be possible after repeatedly low rewards. RAO* applies chance constraints over policies for Partially Observable Markov Decision Processes (Santana, Thiébaux, and Williams 2016), but requires well informed heuristics to converge on a policy quickly. Our approach differs in that we reason over a more general notion of a chance constraint through a risk bounding function, do not explicitly branch on measurements, and use MCTS to produce a policy under limited planning time without heuristics.

A different approach for chance constrained planning is to maximize a sum of reward and a weighted penalty for failure in an unconstrained problem. There is no known method of selecting weights to guarantee chance constraint satisfaction, so the unconstrained problem is solved repeatedly with different weights until the solution is observed to satisfy the chance constraint (Geibel and Wysotzki 2005). We consider large problems for which it is intractable to produce a full policy and calculate the probability of failure. Instead, we define sequential approximating problems that guarantee chance constraint satisfaction without needing to explicitly compute reward or probability of failure for the full policy.

Chance Constrained Markov Decision Process Formulation

We consider a mobile agent with uncertain position taking measurements in a Gaussian Process with known mean and covariance kernel. The vehicle is tasked with maximizing the expected sum of n measurements, while bounding the probability it collides with the environment. We frame this problem as a Chance Constrained Markov Decision Process (CCMDP) with a risk bounding function (Rossman 1977; Ayton and Williams 2018), and we summarize the construction in this section.

CCMDP Definition

A CCMDP with a risk bounding function is defined as a tuple $\langle \mathcal{S}, \mathcal{C}, \mathcal{A}, T, R, s_0, n, \Delta \rangle$, where $\mathcal{S}$ is a set of states, $\mathcal{C} \subseteq \mathcal{S}$ is a set of safe states, $\mathcal{A}$ is a set of actions available from each state, $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ gives the probability of transitioning from a state to another by taking an action, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, \infty)$ gives the reward of entering a state from another by taking an action, $s_0 \in \mathcal{C}$ is a starting state, n is the number of actions that can be taken, and $\Delta\ [0, \infty) \to [0, 1]$ is a risk bounding function which specifies the allowable probability of failure as a function of expected reward. Define a state history as an ordered sequence of states and actions,

$$h_{0:t} = \langle s_0, a_0, s_1, a_1, \ldots, s_t \rangle . \tag{1}$$

The objective is to find a policy $\pi^* : \mathcal{S} \to \mathcal{A}$ defined by

$$\begin{aligned} \pi^* &= \underset{\pi}{\operatorname{argmax}}\ \mathbb{E}\left[g(H_{0:n}) | s_0, \pi\right] \\ \text{s.t. } p&\left(\bigvee_{k=1}^{n} S_k \notin \mathcal{C}\right) \leq \Delta\left(\mathbb{E}\left[g(H_{0:n}) | s_0, \pi\right]\right), \end{aligned} \tag{2}$$

where

$$g(h_{0:n}) = \sum_{k=0}^{n-1} R(s_k, a_k, s_{k+1}). \tag{3}$$

Our use of capital letters emphasizes where variables are considered random.

State Formulation

The agent is unable to detect its position exactly, but models its location at the time of measurement t by a Gaussian distribution with known parameters, $X_t \sim \mathcal{N}(\mu_t, \Sigma_t)$. After each measurement, an action $a_t \in \mathcal{A}$ is chosen which

moves the agent a fixed amount $d(a_t)$, while it is subject to unbounded noise $W_k \sim \mathcal{N}(0, \Sigma_w)$. We assume actions apply identically to each location, so that $X_{t+1} \sim \mathcal{N}(\mu_t + d(a_t), \Sigma_t + \Sigma_w)$. After movement, a measurement $y_{t+1} \in \mathbb{R}$ is taken.

Since the agent is never aware of or capable of responding to its true position history, a state s_t includes a history of position distributions. These are characterized by vectors of mean positions $\boldsymbol{\mu}_{0:t}$ and covariances $\boldsymbol{\Sigma}_{0:t}$. In addition, the agent responds to its entire history of observations $\mathbf{y}_{0:t}$, and whether a failure has occurred is indicated with a binary variable F_t that is zero when no failure has occurred and one otherwise,

$$s_t = \langle \boldsymbol{\mu}_{0:t}, \boldsymbol{\Sigma}_{0:t}, \mathbf{y}_{0:t}, F_t \rangle . \tag{4}$$

The initial state defines the initial mean, covariance, and prior measurements, and is assumed to be safe. $\mathcal{C}$ is the set of all states where $F_t = 0$.

Reward Function

Our problem tasks an agent with maximizing the sum of its measurements. This objective directs the agent to confirm the presence of high value measurements, where value increases with the magnitude of the measurement.

We model the agent as receiving reward immediately after measurement. Upon collision with the environment, which is described by a forbidden region $\mathcal{F}$, it is unable to perform the measurement, but previous rewards are not lost. For an underwater vehicle, this case occurs when collision triggers a mission abort and surface sequence, which takes the vehicle out the field until a diagnostic can be run and parts can be replaced, but previous measurements are recoverable. Successful abort sequences after a collision are common for slow vehicles with line of sight to the surface, and we leave the less common case where data is lost in collision to future research.

Since arbitrarily low measurements are possible, we specify a minimum reward R_{min} for each action. We then scale the problem such that the minimum reward is 0, and provide 0 reward for failure. Intuitively, this specifies that all measurements below a certain threshold are uninteresting.

We are interested in sampling high reward locations for the purpose of confirming where high measurement values reside, and there is little benefit to repeatedly measuring a global maximum. We therefore define a threshold distance l_{min} and impose that a measurement within l_{min} of a previous measurement is worth zero reward.

Altogether, this specifies a reward for measurements $\mathbf{y}_{i:t}$ with mean location history $\boldsymbol{\mu}_{0:t}$ of

$$\tilde{R}(\boldsymbol{\mu}_{0:t}, \mathbf{y}_{i:t}, \mathbf{F}_{i:t}) = \sum_{k=i}^{t} \max(0,\ y_k - R_{min})\ \mathbb{1}(\boldsymbol{\mu}_{0:k}, F_k), \tag{5}$$

where

$$\mathbb{1}(\boldsymbol{\mu}_{0:t}, F_t) = \begin{cases} 1 & F_t = 0 \text{ and } \bigwedge_{k=0}^{t-1} \|\mu_t - \mu_k\| \geq l_{min} \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

$\tilde{R}$ is introduced for later convenience, while the CCMDP reward function is

$$R(s_{t-1}, a_{t-1}, s_t) = \tilde{R}(\boldsymbol{\mu}_{0:t}, y_t, F_t). \tag{7}$$

State Transitions

We model the environment which the agent measures as a Gaussian Process. Gaussian Processes have been frequently applied for informative planning because they can be fully specified with intuitive mean and covariance models, and due to their capability to model a large class of functions, output probability distributions over measurements, and support exact inference in polynomial time (Williams and Rasmussen 2006).

A GP is fully specified by a mean function $m(x)$ and a covariance kernel $k(x, x')$, which specify the mean value at a point and the covariance between two points respectively. Let $\mathbf{x}_{0:t}$ be the vector of past locations with corresponding measurements $\mathbf{y}_{0:t}$, and $\mathbf{x}_*$ be a vector of prediction locations. Define $M(\mathbf{x})$ as the vector $[M(\mathbf{x})]_i = m(x_i)$ and $K(\mathbf{x}, \mathbf{x}')$ as the matrix $[K(\mathbf{x}, \mathbf{x}')]_{ij} = k(\mathbf{x}_i, \mathbf{x}'_j)$. Then the posterior probability distribution of the predicted measurements is a Gaussian,

$$\mathbf{y}_* | \mathbf{y}_{0:t}, \mathbf{x}_{0:t}, \mathbf{x}_* \sim \mathcal{N}\left(\theta^t_*, \kappa^t_*\right), \tag{8}$$

where

$$\theta^t_* = M(\mathbf{x}_*) + K(\mathbf{x}_*, \mathbf{x}_{0:t}) K^{-1}_{yy} \left(\mathbf{y}_{0:t} - M(\mathbf{x}_{0:t})\right), \tag{9}$$

$$\kappa^t_* = K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{x}_{0:t}) K^{-1}_{yy} K(\mathbf{x}_{0:t}, \mathbf{x}_*) + \sigma^2 I, \tag{10}$$

$$K_{yy} = K(\mathbf{x}_{0:t}, \mathbf{x}_{0:t}) + \sigma^2 I, \tag{11}$$

and σ^2 is the variance in measurements (Williams and Rasmussen 2006). We draw attention to our use of a superscript t to denote predictions based on measurements $\mathbf{y}_{0:t}$.

Though measurement predictions can be done exactly, it is difficult to compute the true probability of collision from an action. X_t is described by a Gaussian distribution, but when conditioned on the safety of previous states it is not Gaussian in general, since this suggests that $X_k \notin \mathcal{F}$ for all $k \leq t$. In our approximating CCMDPs, we will instead conservatively overestimate the value.

For appropriate mean and covariances, and from a safe state to another it follows that

$$T(s_t, a_t, s_{t+1}) = p\left(F_{t+1} = 0 | s_t, a_t\right) \times \int p(y_{t+1} | \mathbf{y}_{0:t}, \mathbf{x}_{0:t+1})\ p(\mathbf{x}_{0:t+1} | \mathbf{F}_{0:t+1} = \mathbf{0})\ d\mathbf{x}_{0:t+1}. \tag{12}$$

It is useful to think of failure states as being terminal, with no actions available from them. Formally, we define a single action as available which always leads back to the same state, netting zero reward.

Constraining Failure Probability

An unconstrained MDP considers failure in the sense that the expected reward decreases as failure probability increases. However, the optimal unconstrained policy may result in a probability of failure that is arbitrarily close to 1,

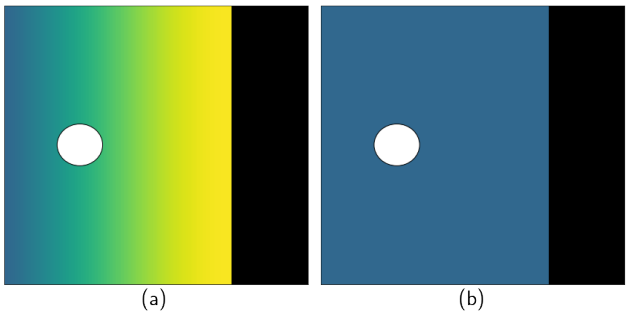

Figure 1: An agent (white) close to an obstacle (black) with (a) high reward close to the obstacle and (b) uniform reward.

while simultaneously placing no constraint on how large the reward must be in this case. This behavior is undesirable for a mission designer, for whom the expense of collecting and repairing the vehicle may not be worth the data obtained.

We encode the tolerance for risk of a mission designer through the risk bounding function, which specifies the allowable probability of failure of a policy as a function of the expected reward. We require that Δ is a non-decreasing concave function, which encodes that we will not allow less risk for higher reward, and that progressively additional reward is worth less additional risk.

Risk bounding functions generalize the notion of a single risk bound that appears in the chance constrained planning literature (Ono and Williams 2008). The idea of a functional representation of failure tolerance is particularly important in exploration, where the increase in reward with risk is unknown. This idea is illustrated in Figure 1, where in (a) a small increase in risk allows the agent to move closer to the obstacle and leads to a large increase in reward, while in (b) it has no effect. A mission designer may prefer a slightly higher risk bound in (a), but a lower risk bound in (b). A risk bounding function encodes this tolerance without the need to know the relationship between risk and reward in the environment, only the price in failure probability that the mission designer is willing to pay for reward.

It is important to note that the chance constraint uses expectations of failure and reward over the entire policy. Dangerous actions with low reward are permitted if high reward is achieved on other measurement histories. This is important, beause an unlikely environment may result in minimum rewards for all actions, but it may not be possible to execute actions for which the probability of failure is below $\Delta(0)$.

Problem Statement

The chance constrained measurement maximizing adaptive sampling problem is summarized in Problem 1.

Problem 1. *Chance Constrained Measurement Maximizing Adaptive Sampling*

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\ \mathbb{E}\left[g(H_{0:n})|\pi\right]$$
$$\text{s.t. } p\left(\bigvee_{k=1}^{n} X_k \in \mathcal{F}\right) \leq \Delta\left(\mathbb{E}\left[g(H_{0:n})|\pi\right]\right)$$
$$Y \sim \mathcal{GP}\left(m, k\right)$$
$$X_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$$
$$X_{t+1} \sim \mathcal{N}(\mu_t + d(a_t), \Sigma_t + \Sigma_w)$$
$$g(h_{0:n}) = \tilde{R}(\boldsymbol{\mu}_{0:n}, \mathbf{y}_{1:n}, \mathbf{F}_{1:n})$$

Our approach to Problem 1 is to formulate a sequence of approximating CCMDPs. We show how a policy guaranteed to satisfy the failure probability constraints of Problem 1 may be constructed by placing constraints on the approximating CCMDPs, and we use Monte Carlo Tree Search methods to ensure a policy is available at any time.

Enforcing Chance Constraints Using Vulcan

Problem 1 is intractable even if the measurements are discretized because the state space grows exponentially in the number of actions and measurements and the transition probabilities in (12) are expensive to compute. Instead, our strategy is to use Monte Carlo Tree Search (MCTS) techniques on approximating CCMDPs so (12) does not need to be computed exactly and an approximately optimal policy is found in an anytime manner. In order to use MCTS to guarantee a policy satisfies a risk bounding function, we use the *Vulcan* algorithm (Ayton and Williams 2018).

The Vulcan Algorithm

Vulcan is based on the *Upper Confidence Bound applied to Trees* (UCT) algorithm for MDPs, which uses previous samples of the search tree to guide future samples towards promising solutions (Kocsis and Szepesvári 2006). In UCT, random sampling is performed to build a search tree. At each sampled state, each action is sampled once, and on subsequent samples the action is chosen according to

$$a_t = \underset{a}{\operatorname{argmax}}\ Q(s_t, a) + \sqrt{\frac{2\log N_{s_t}}{N_{s_t,a}}}. \tag{13}$$

$Q(s_t, a_t)$ is an estimate of the maximum expected reward to go by taking action a_t based on samples, N_{s_t} is the number of samples taken at state s_t, and N_{s_t,a_t} is the number of samples of a_t from s_t.

To apply UCT to CCMDPs, Vulcan defines the sequence execution risk *ser* of a state history $h_{0:t} = \langle s_0, a_0, s_1, a_1, \ldots, s_t\rangle$ as

$$ser(h_{0:n}) = \begin{cases} \frac{p\left(\bigvee_{k=1}^{n} S_k \notin \mathcal{C} \middle| s_0, \mathbf{a}_{0:n-1}\right)}{1-p\left(\bigvee_{k=1}^{n} S_k \notin \mathcal{C} \middle| s_0, \mathbf{a}_{0:n-1}\right)} & \text{no failures} \\ 0 & \text{otherwise} \end{cases}. \tag{14}$$

UCT proceeds as in the work of Kocsis and Szepesvári (2006), but upon reaching a safe state at the planning horizon n, the entire state history to that state is checked to satisfy

$$ser(h_{0:n}) \leq \Delta\left(f(h_{0:n})\right), \tag{15}$$

where f is any function that satisfies $\mathbb{E}\left[f(H_{0:n})|s_0,\pi\right] \leq \mathbb{E}\left[g(H_{0:n})|s_0,\pi\right]$. By the definition of *ser*, (15) is satisfied for any histories ending in failure states, which is desirable so that risks with low immediate rewards are allowed if they lead to high rewards later. Consequently, it is not necessary to consider failure state histories explicitly. If (15) is not satisfied or no actions remain from a state, then the last action is deleted from the search tree and a new sample is taken from s_0.

Theorem 1 of Ayton and Williams (2018) shows that any policy found under this strategy is guaranteed to satisfy the chance constraint. The proof follows from the fact that the expectation of *ser* across all state histories in a policy equals the total probability of failure, so that

$$\begin{aligned} p\left(\bigvee_{k=1}^{n} S_k \notin \mathcal{C} \middle| s_0,\pi\right) &= \mathbb{E}\left[ser\left(H_{0:n}\right)|s_0,\pi\right] \\ &\leq \mathbb{E}\left[\Delta\left(f(H_{0:n})\right)|s_0,\pi\right] \\ &\leq \Delta\left(\mathbb{E}\left[f(H_{0:n})|s_0,\pi\right]\right) \\ &\leq \Delta\left(\mathbb{E}\left[g(H_{0:n})|s_0,\pi\right]\right) \end{aligned} \tag{16}$$

by Jensen's inequality for non-decreasing concave Δ.

The resultant policy is suboptimal, but the advantage in this context is that (15) can be applied to each state history without knowledge of the others. A policy is guaranteed to satisfy the chance constraint if a state history satisfying (15) *can be* computed after any set of observations, *even if they are not computed explicitly*.

Risk Approximations

Since probability of failure is difficult to compute exactly, we follow Ono and Williams (2008) to develop a conservative bound using Boole's inequality.

$$\begin{aligned} p\left(\bigvee_{k=1}^{n} S_k \notin \mathcal{C}|s_0,\mathbf{a}_{0:n-1}\right) &= p\left(\bigvee_{k=1}^{n} X_k \in \mathcal{F}|\mu_k,\Sigma_k\right) \\ &\leq \sum_{k=1}^{n} p\left(X_k \in \mathcal{F}|\mu_k,\Sigma_k\right), \end{aligned} \tag{17}$$

where

$$p(X_k \in \mathcal{F}|\mu_k,\Sigma_k) = \int_{\mathcal{F}} \mathcal{N}\left(\mu_k,\Sigma_k\right) dx_k. \tag{18}$$

(18) remains computationally intensive for arbitrary obstacles, and sampling based approximations can underestimate the risk. Instead, we use an estimation that is guaranteed to be conservative by enclosing $\mathcal{F}$ with a union of N_F convex polytopes,

$$\mathcal{F} \subseteq \bigcup_{i=1}^{N_F} \mathcal{F}_i, \tag{19}$$

where polytope $\mathcal{F}_i$ may be described as an intersection of half-spaces based on each of its E_i edges,

$$\mathcal{F}_i = \left\{ x \;\middle|\; \bigwedge_{j=1}^{E_i} h_{ij}^T x \geq g_{ij} \right\}, \tag{20}$$

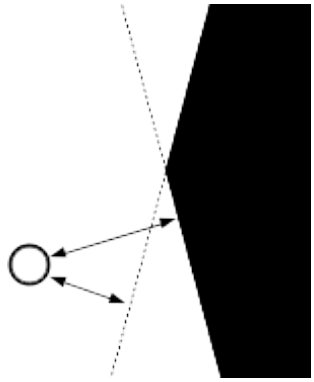

Figure 2: Probability of collision with a convex obstacle is less than the minimum probability of crossing a line defined by its edges.

for vector h_{ij} and scalar g_{ij}.

Using Boole's inequality again, we have

$$p(X_k \in \mathcal{F}|\mu_k,\Sigma_k) \leq \sum_{i=1}^{N_F} p(X_k \in \mathcal{F}_i|\mu_k,\Sigma_k). \tag{21}$$

Let $\mathcal{E}_i(\mu_k)$ be the set of half-space indices of $\mathcal{F}_i$ for which the mean state lies outside,

$$\mathcal{E}_i(\mu_k) = \left\{ j \;\middle|\; h_{ij}^T \mu_k < g_{ij} \right\}. \tag{22}$$

Then we bound the probability of collision by the minimum probability of entering one of the half-spaces in $\mathcal{E}_i(\mu_k)$:

$$p(X_k \in \mathcal{F}_i|\mu_k,\Sigma_k) \leq \min_{j \in \mathcal{E}_i(\mu_k)} p(h_{ij}^T X_k \geq g_{ij}), \tag{23}$$

$$p(h_{ij}^T X_k \geq g_{ij}) = \frac{1}{2} + \frac{1}{2}\text{erf}\left(\frac{h_{ij}^T \mu_k - g_{ij}}{\sqrt{2h_{ij}^T \Sigma_k h_{ij}}}\right). \tag{24}$$

The intuition for (23) is given in Figure 2, while (24) follows from a Gaussian CDF. A convex decomposition of the forbidden regions can be computed prior to the start of planning.

For convenience, we define

$$p_f(s_k) = \sum_{i=1}^{N_F} \min_{j \in \mathcal{E}_i(\mu_k)} p(h_{ij}^T X_k \geq g_{ij}) \tag{25}$$

Then it follows from (14), (17), and (21) that

$$ser(h_{0:n}) \leq \frac{\sum_{k=1}^{n} p_f(s_k)}{1 - \sum_{k=1}^{n} p_f(s_k)}. \tag{26}$$

Our approach is then to sample over potential trajectories, and verify that each potential trajectory satisfies

$$\frac{\sum_{k=1}^{n} p_f(s_k)}{1 - \sum_{k=1}^{n} p_f(s_k)} \leq \Delta\left(f(h_{0:n})\right). \tag{27}$$

Solution Procedure

Even using Vulcan, Problem 1 is too large to find a reasonable solution, as Vulcan requires every outcome of the policy to be sampled in order to guarantee the chance constraint is satisfied. Instead, when planning from state s_t, we construct an approximating CCMDP starting at s_t. Action a_k results in failure with probability $p_f(s_{k+1})$ or a single success state, and provides reward

$$R^{ap}(s_{k-1}, a_{k-1}, s_k) = \tilde{R}(\boldsymbol{\mu}_{0:k}, \theta_k^t, F_k). \tag{28}$$

θ_k^t is computed using (8) through (11) as if all observations are taken at $\boldsymbol{\mu}_{0:k}$. This approximation produces a small amount of error under the reasonable assumption that position distribution length scales are small compared to GP kernel length scales. Since there is one safe outcome for each action, a single state history constitutes a valid policy which does not depend on measurement outcomes. The first action of the policy is executed, then replanning is performed from the true s_{t+1}.

Chance Constraint Satisfaction

The key to enforcing the chance constraint is viewing the strategy above as being applied to every possible sequence of states in the true CCMDP. After t actions, an approximating CCMDP exists for each s_t, and a policy is defined by the actions that were selected up to s_t and the actions found by the approximating CCMDPs. By requiring that all approximating CCMDP solutions would result in state histories that satisfy (27), we guarantee the chance constraint without needing to explicitly solve all the approximating problems.

When solving the approximating CCMDPs, we require that safe state histories satisfy

$$\begin{aligned}\frac{\sum_{k=1}^n p_f(s_k)}{1-\sum_{k=1}^n p_f(s_k)} \leq \Delta\Bigg(&\sum_{k=1}^{t} \tilde{R}(\boldsymbol{\mu}_{0:k}, \theta_k^{k-1}, F_k) \\ &+ \tilde{R}(\boldsymbol{\mu}_{0:n}, \boldsymbol{\theta}_{t+1:n}^t, \mathbf{F}_{t+1:n})\Bigg).\end{aligned} \tag{29}$$

The true observations $\mathbf{y}_{0:t}$ are used in the computation of the means θ, but their values do not enter the constraint directly. This is a method to ensure that a policy exists satisfying the above even if all measurements are consistently low.

By finding a policy satisfying (29) from every reachable state s_t, (27) is satisfied with

$$\begin{aligned}f(h_{0:n}) = \sum_{k=1}^{t} &\tilde{R}\left(\boldsymbol{\mu}_{0:k}, \mathbb{E}\left[Y_k|\mathbf{y}_{0:k-1}\right], F_k\right) \\ &+ \tilde{R}\left(\boldsymbol{\mu}_{0:n}, \mathbb{E}\left[\mathbf{Y}_{t+1:n}|\mathbf{y}_{0:t}\right], \mathbf{F}_{t+1:n}\right).\end{aligned} \tag{30}$$

It follows from Jensen's inequality and the convexity of the max function that

$$\begin{aligned}\tilde{R}(\boldsymbol{\mu}_{0:k}, \theta_k^t, F_k) &= \tilde{R}(\boldsymbol{\mu}_{0:k}, \mathbb{E}\left[Y_k|\mathbf{y}_{0:t}\right], F_k) \\ &\leq \mathbb{E}\left[\tilde{R}(\boldsymbol{\mu}_{0:k}, Y_k, F_k)\middle|\mathbf{y}_{0:t}\right],\end{aligned} \tag{31}$$

so $\mathbb{E}\left[f(H_{0:n})|s_0, \pi\right] \leq \mathbb{E}\left[g(H_{0:n})|s_0, \pi\right]$ and the chance constraint is satisfied. It is *not* necessarily true that the probability of failure is bounded by the risk bounding function applied to the observations that actually occur, as desired.

Guarantees on Existence of Policy

The above strategy guarantees that a policy can be can be found that follows the risk bounding function, assuming that a policy that satisfies (29) can be computed in response to all measurements. This is a non-trivial assertion, and in this section we introduce an additional condition that ensures this is always possible.

Consider planning from state s_t, with a potential policy that satisfies (29) and includes a_t. Define a *worst case state history* $w(a_t)$ as any safe state history $w(a_t) = \left\langle s_0^{w(a_t)}, a_0^{w(a_t)}, s_1^{w(a_t)}, \ldots, s_n^{w(a_t)} \right\rangle$ that satisfies $s_{0:t}^{w(a_t)} = s_{0:t}$, $a_{0:t}^{w(a_t)} = a_{0:t}$, and

$$\frac{\sum_{k=1}^n p_f(s_k^{w(a_t)})}{1-\sum_{k=1}^n p_f(s_k^{w(a_t)})} \leq \Delta\left(\sum_{k=1}^{t+1} \tilde{R}(\boldsymbol{\mu}_{0:k}, \theta_k^{k-1}, F_k)\right). \tag{32}$$

By requiring that any worst case state history exists for the chosen a_0, the capability to replan according to the stategy in this paper is guaranteed.

To see this, assume that when planning from state s_t that a policy is found that satisfies (29) and $w(a_t)$ exists. When replanning from any possible s_{t+1}, following the actions $a_{t+1:n-1}^{w(a_t)}$ will always satisfy (29) as well, even if all measurements result in predicted means below the minimum reward. Following $a_{t+1:n-1}^{w(a_t)}$ from any s_{t+1} also produces a worst case state history for $a_{t+1}^{w(a_t)}$, which means that the worst case state history is a policy that can be followed until the end of execution, guaranteeing that (29) and (32) can always be satisfied regardless of measurements.

When replanning from s_{t+1}, usually $a_{t+1} \neq a_{t+1}^{w(a_t)}$. In this case, a_{t+1} is only permitted if $w(a_{t+1})$ can be found, which guarantees the risk bounding function can be satisfied regardless of future measurements by the same argument.

To summarize, the worst case state history is typically not executed, but one must exist. This is a weaker condition than enforcing that the solution to the approximating CCMDP must be a worst case state history, or imposing a probability of failure constraint on every path based on its measurements. Furthermore, the existence of a worst case state history implies a worst case state history exists when replanning from the next state.

Practically, the existence of a worst case state history can be checked after sampling a_t when planning from s_t. If one cannot be found, the action is immediately deleted from the search space. Worst case state histories can often be found by greedily selecting the minimum risk action from the action space. In our implementation, if this is not a valid worst case state history, none is assumed to exist.

Algorithm Description

Our strategy requires solving n approximating CCMDPs, which we describe in Algorithm 1. Each approximating CCMDP is solved by repeatedly sampling from the current state history up to a time limit of τ. At the time bound, the agent executes the first action of the best found policy so far.

Algorithm 1: ExecuteRiskBoundedPolicy

Input: Initial state s_0, planning time limit τ

$h_0 \leftarrow s_0$
for t from 0 to $n-1$ **do**
  while sample time $< \tau$ **do**
    Sample($h_{0:t}$)
  execute $a_t = \text{argmax}_a\, Q(s_t, a)$
  $s_{t+1} \leftarrow$ next state with measurement y_{t+1}
  reset search tree

Algorithm 2 describes the sampling strategy for the approximating CCMDPs. A search tree is built from s_t according to the action selection rules of the UCT algorithm. Vulcan guarantees that it is only ever necessary to sample from safe states, so each action leads to a single state deterministically.

Upon reaching the planning horizon, (29) is verified on line 2. After selecting action a_t, $w(a_t)$ is found on line 7 and (32) is verified. If either of these conditions fail, or no actions exist at a non-terminal state at line 4, the immediately preceding action is deleted, and sampling restarts from the root node. This ensures that the highest reward policy found satisfies the chance constraint.

Algorithm 2: Sample

Input: State history $h_{0:t}$

1 **for** k from t to $n-1$ **do**
2 **if** $k = n$ and (29) not satisfied **then**
3 delete a_{n-1} and **return**
4 **if** no actions at s_k **then**
5 delete a_{k-1} and **return**
6 **if** $k = t+1$ **then**
7 $w \leftarrow$ greedily found sequence of min risk states
8 **if** (32) not satisfied **then**
9 delete a_t and **return**
10 **if** unsampled action at s_k **then**
11 $a_k \leftarrow$ unsampled action
12 **else**
13 $a_k \leftarrow \text{argmax}_a\, Q(s_k, a) + \sqrt{\frac{2\log(N_{s_k})}{N_{s_k,a_k}}}$
14 $s_{k+1} \leftarrow$ next state with measurement θ^t_{k+1}
15 **for** k from $n-1$ to t **do**
16 $Q(s_k, a_k) \leftarrow (1 - p_f(s_{k+1}))\,(R^{ap}(s_k, a_k, s_{k+1}) + \max_a Q(s_{k+1}, a))$
17 increment N_{s_k}, N_{s_k,a_k}

Experiments

We examine our algorithm in two different ways. First, we run the algorithm on real bathymetry data and a simulated measurement field. We show our algorithm is able to move towards high reward locations based on the data it gathers, and take dangerous actions when they are expected to yield high reward. We then verify that the risk bounding function is satisfied through Monte Carlo simulations over randomly instantiated Gaussian Processes. To the best of our knowledge, no other algorithm is capable of performing chance constrained adaptive sampling on the scale we consider.

Tests on Controlled Environments

To test the performance of our algorithm in realistic scenarios, we convexify true bathymetric data to produce forbidden regions, and simulate measurement fields. The location was East of Boston Harbor, from -70.890 to -70.876 degrees longitude, and 42.344 to 42.355 degrees latitude, provided by NOAA survey H10992 (National Oceanic and Atmospheric Administration 2001). The mission simulated an autonomous underwater vehicle operating at a constant 15 meters depth and maximizing temperature measurements, so 15 meter depth contours were used as obstacle boundaries. In each case, the agent started at a location -70.8816 degrees longitude and 42.3505 degrees latitude with zero position uncertainty. The Gaussian Process covariance kernel and the vehicle position covariance were chosen to be indicative of a true temperature measurement mission. The true value of the measured field was 16 at the starting location, and increased by 1 for each km West or South. Each action moved the vehicle 50 meters in one of the eight compass directions. We used the following parameters: $n = 20$, $\tau = 60$ sec, $\Sigma_0 = 0I$ m^2, $\Sigma_w = 12I$ m^2, $R_{min} = 12.5$, $l_{min} = 12.5$ m, $m(x) = 16$, $k(x, x') = 1.25 \exp\left(-\|x - x'\|^2 / (2 \times (200\text{ m})^2)\right)$, $\sigma = 0$.

In Figure 3 we test this scenario using three different risk bounding functions. The risk bounding functions were selected primarily to show differences in behavior, but we note that they lead to realistic acceptable failure rates on the order of tenths of a percent. Figure 3 (a) shows a trajectory resulting from a risk bounding function $\Delta(x) = 0.0003x$. The measurements are high enough to warrant movement into the South-West of the map by the most direct route possible, which requires passing between multiple obstacles. In Figure 3 (b) a lower risk bounding function of $\Delta(x) = 0.0002x$ does not allow movement close to obstacles until there is enough certainty that high measurements lie to the South-West. In addition, the route taken uses a thicker channel, with less overall probability of failure. Finally, in Figure 3 (c), the risk bounding function $\Delta(x) = 0.0001x$ is too strict to allow the vehicle to pass close to obstacles, and instead it moves up and down the border of the obstacles without moving in too close.

Monte Carlo Tests

In order to experimentally verify that the risk bounding function is satisfied across a policy, we ran Monte Carlo simulations with random measurements following a known Gaussian Process, and verified that the failure rate was less than the risk bounding function applied to the average reward. In the simulations, true (disturbed) locations were generated, and measurements were drawn from a Gaussian Process at

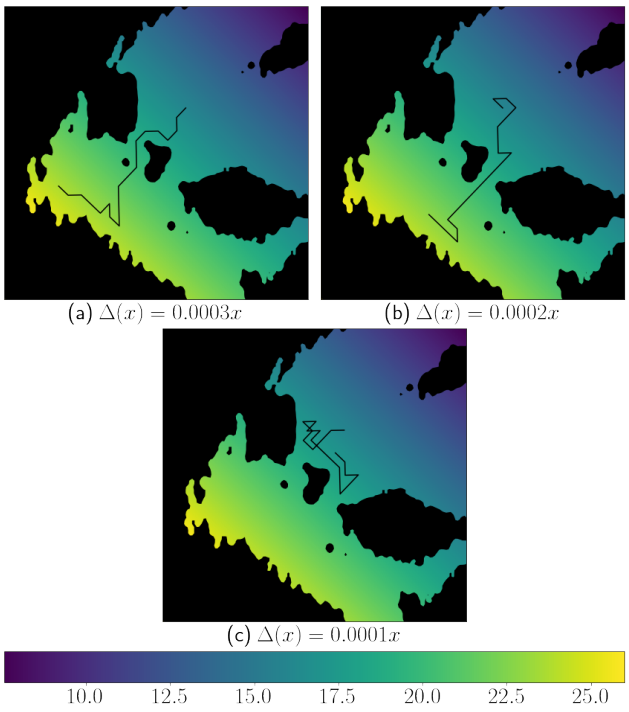

Figure 3: Output trajectories with a single true environment and multiple risk bounding functions.

the true locations, while the algorithm reasoned over measurements at mean locations. The environment was the same as the previous test.

The low probability of failure in the previous test meant that uncertainty in a Monte Carlo derived failure rate would be comparable to the true failure rate. To increase certainty in the simulation results, we increased the covariance of the agent, leading to a higher probability of failure. In order to speed up simulations and emphasize risk, we also decreased the planning horizon and planning time, and allowed the agent to move further with every action. The following parameters were changed: $n = 8$, $\tau = 2$ sec, $\Sigma_w = 60I$ m^2, $l_{min} = 25$ m. The distance traveled by every action was changed to 100 m. Failure was evaluated with respect to the convexified obstacles, so the failure rate does not account for conservatism due to convexification. 10,000 simulations were run with a risk bounding function of $\Delta(x) = 0.001x$. The mean function was the true environment of the previous experiment so that measurements would typically be biased towards dangerous actions.

The expected cumulative reward was 64.9, which permitted a failure rate of 0.0649 under the risk bounding function, while the measured failure rate was 0.0208. There was conservatism in the policy, as only 32% of permitted risk was used. The conservatism can be attributed to three major sources. First and most importantly, our strategy averages reward and risk across outcomes, but does not move all allowed probability of failure from low risk to high risk outcomes. In particular, some environments resulting from the GP have high reward to the East where there are few obstacles. Our approach is not fully capable of moving all allowed risk to cases where high rewards are near obstacles. The additional sources of conservatism are the use of Boole's inequality to overestimate the probability of failure and the underestimation of the reward function.

To confirm that conservatism was reduced when danger exists in all directions, we reran the experiment with $\Sigma_w = 100I$ m^2 and additional obstacles introduced to the North and East. In this case the expected cumulative reward was 26.4, which permitted a failure rate of 0.0264, while the measured failure rate was 0.0165. In this case, 63% of available risk was used.

Summary

In this paper, we developed a method of finding an adaptive policy where the probability of failure is bounded as a convex function of expected reward. We derived constraints that enforce the chance constraint without the need to plan over all outcomes and which guarantee replanning is possible. By applying Monte Carlo Tree Search to a series of easily computable approximating problems, we ensure that an action is found in an anytime manner. Simulation results on true bathymetry show our algorithm trades off risk against reward intuitively, taking dangerous actions only when justified by the reward, while Monte Carlo simulations verify that the chance constraint is satisfied.

Acknowledgements

This work was funded by the Exxon Mobil Corporation under the MIT Energy Initiative (grant EM09079). We would also like to thank Siyu Dai for her comments on the initial draft of this paper, and the three anonymous reviewers who improved the clarity of this work.

References

Ayton, B., and Williams, B. 2018. Vulcan: a monte carlo algorithm for large chance constrained mdps with risk bounding functions. https://arxiv.org/abs/1809.01220.

Binney, J.; Krause, A.; and Sukhatme, G. S. 2010. Informative path planning for an autonomous underwater vehicle. In *IEEE International Conference on Robotics and Automation*, 4791–4796.

Blackmore, L.; Ono, M.; Bektassov, A.; and Williams, B. C. 2010. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE transactions on Robotics* 26(3):502–517.

Bryan, B.; Nichol, R. C.; Genovese, C. R.; Schneider, J.; Miller, C. J.; and Wasserman, L. 2006. Active learning for identifying function threshold boundaries. In *Advances in neural information processing systems*, 163–170.

Garnett, R.; Krishnamurthy, Y.; Xiong, X.; Schneider, J.; and Mann, R. 2012. Bayesian optimal active search and surveying. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 843–850. Omnipress.

Geibel, P., and Wysotzki, F. 2005. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research* 24:81–108.

Gotovos, A.; Casati, N.; Hitz, G.; and Krause, A. 2013. Active learning for level set estimation. In *IJCAI*, 1344–1350.

Hitz, G.; Galceran, E.; Garneau, M.-È.; Pomerleau, F.; and Siegwart, R. 2017. Adaptive continuous-space informative path planning for online environmental monitoring. *Journal of Field Robotics* 34(8):1427–1449.

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

Kocsis, L., and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *European conference on machine learning*, 282–293. Springer.

Krause, A., and Guestrin, C. 2007. Near-optimal observation selection using submodular functions. In *AAAI*, volume 7, 1650–1654.

Krause, A.; Singh, A.; and Guestrin, C. 2008. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research* 9(Feb):235–284.

Low, K. H.; Dolan, J. M.; and Khosla, P. K. 2009. Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. In *ICAPS*, 233–240.

Marchant, R.; Ramos, F.; Sanner, S.; et al. 2014. Sequential bayesian optimisation for spatial-temporal monitoring. In *UAI*, 553–562.

National Oceanic and Atmospheric Administration. 2001. Survey H10992. https://www.ngdc.noaa.gov/nos/H10001-H12000/H10992.html.

Ono, M., and Williams, B. C. 2008. An efficient motion planning algorithm for stochastic dynamic systems with constraints on probability of failure. In *AAAI*, 1376–1382.

Rossman, L. A. 1977. Reliability-constrained dynamic programing and randomized release rules in reservoir management. *Water Resources Research* 13(2):247–255.

Santana, P.; Thiébaux, S.; and Williams, B. 2016. Rao*: an algorithm for chance constrained pomdps. In *Proc. AAAI Conference on Artificial Intelligence*.

Singh, A.; Nowak, R.; and Ramanathan, P. 2006. Active learning for adaptive mobile sensing networks. In *Proceedings of the 5th international conference on Information processing in sensor networks*, 60–68. ACM.

Williams, C. K., and Rasmussen, C. E. 2006. Gaussian processes for machine learning. *the MIT Press* 2(3):4.

Yilmaz, N. K.; Evangelinos, C.; Lermusiaux, P. F.; and Patrikalakis, N. M. 2008. Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming. *IEEE Journal of Oceanic Engineering* 33(4):522–537.