Bounded Policy Synthesis for POMDPs with Safe-Reachability Objectives

Robotics Track

Yue Wang
Department of Computer Science
Rice University
Houston, TX 77005, USA
yw27@rice.edu

Swarat Chaudhuri
Department of Computer Science
Rice University
Houston, TX 77005, USA
swarat@rice.edu

Lydia E. Kavraki
Department of Computer Science
Rice University
Houston, TX 77005, USA
kavraki@rice.edu

ABSTRACT
Planning robust executions under uncertainty is a fundamental challenge for building autonomous robots. Partially Observable Markov Decision Processes (POMDPs) provide a standard framework for modeling uncertainty in many applications. In this work, we study POMDPs with safe-reachability objectives, which require that with a probability above some threshold, a goal state is eventually reached while keeping the probability of visiting unsafe states below some threshold. This POMDP formulation is different from the traditional POMDP models with optimality objectives and we show that in some cases, POMDPs with safe-reachability objectives can provide a better guarantee of both safety and reachability than the existing POMDP models through an example. A key algorithmic problem for POMDPs is policy synthesis, which requires reasoning over a vast space of beliefs (probability distributions). To address this challenge, we introduce the notion of a goal-constrained belief space, which only contains beliefs reachable from the initial belief under desired executions that can achieve the given safe-reachability objective. Our method compactly represents this space over a bounded horizon using symbolic constraints, and employs an incremental Satisfiability Modulo Theories (SMT) solver to efficiently search for a valid policy over it. We evaluate our method using a case study involving a partially observable robotic domain with uncertain obstacles. The results show that our method can synthesize policies over large belief spaces with a small number of SMT solver calls by focusing on the goal-constrained belief space.

KEYWORDS
Planning under Uncertainty; Policies; Robotics; Formal Methods

ACM Reference Format:

1 INTRODUCTION
Partially Observable Markov Decision Processes (POMDPs) [38] provide a principled mathematical framework for modeling a variety of problems in the face of uncertainty [5, 11, 22, 29]. As an example, in robotics, accounting for uncertainty is a fundamental challenge for deploying autonomous robots in the physical world. Many applications in uncertain robotic domains can be modeled as POMDP problems [5, 6, 15, 22].

A key algorithmic problem for POMDPs is the synthesis of policies [38]: recipes that specify the actions to take under all possible events in the environment. Typically, the goal in policy synthesis is to find optimal solutions with respect to quantitative objectives such as maximizing discounted reward [1, 12, 15–17, 25, 26, 39, 40], while the purely quantitative formulations of the problem are suitable for many applications, there are, settings that demand synthesis with respect to boolean requirements. For example, consider the scenario shown in Figure 1 where we want to guarantee that a robot can accomplish a task safely in an uncertain domain. This goal is naturally formulated as policy synthesis from a high-level requirement written in a temporal logic. Moreover, in some cases, formulating boolean requirements as quantitative objectives by assigning negative rewards for states that violate the boolean requirements and positive rewards for states that satisfy the boolean requirements, leads to policies that are overly conservative or overly risky [43], depending on the particular reward function chosen. Therefore, new models and algorithms are required for handling POMDPs with boolean requirements explicitly. In Section 4, we discuss an example that shows in some scenarios, handling boolean requirements explicitly in POMDPs provides a better guarantee of both safety and reachability than the traditional quantitative POMDP formulations.

Policy synthesis in POMDPs with respect to boolean requirements has been studied before. Specifically, inspired by applications
in robotics, the qualitative analysis problem of almost-sure satisfaction of POMDPs with temporal logic specifications was first introduced in [6]. In their work, the goal is to find policies that satisfy a temporal property with probability 1.

A more general quantitative analysis problem of POMDPs with temporal logic specifications is to synthesize policies that satisfy a temporal property with a probability above some threshold. In this work, we study this problem for the special case of safe-reachability properties, which require that with a probability above some threshold, a goal state is eventually reached while keeping the probability of visiting unsafe states below some threshold. Many robot tasks such as the one in Figure 1 can be formulated using a safe-reachability objective.

Previous results [7, 27, 33] have shown that the quantitative analysis problem of POMDPs with reachability objectives is undecidable. To make the problem tractable, we assume there exists a bounded horizon $h$ such that $h$ is sufficiently large to prove the existence of a valid policy, or the user is not interested in plans beyond the bounded horizon $h$. This assumption is particularly reasonable for robotic domains because robots are often required to accomplish a task in bounded steps due to some resource constraints such as energy/time constraints. Figure 1 shows an example of such a scenario: a robot with uncertain actuation and perception needs to navigate through a kitchen to pick up an object in bounded steps while avoiding collisions with uncertain obstacles.

To the best of our knowledge, the quantitative analysis problem of POMDPs with safe-reachability objectives has not been considered before. In this work, we present a practical policy synthesis approach for this problem. Like most other algorithms for policy synthesis, our approach is based on reasoning about the space of beliefs, or probability distributions over possible states of the POMDP. Our primary algorithmic challenge is that the belief space is a vast, high-dimensional space of probability distributions.

Our approach to this challenge is based on the new notion of a goal-constrained belief space. This notion takes inspiration from recent advances in point-based algorithms [25, 26, 34, 39] for POMDPs with discounted reward objectives. These POMDP algorithms exploit the notion of the reachable belief space $R(b_{init})$ from an initial belief $b_{init}$ and compute an approximately optimal policy over $R(b_{init})$ rather than the entire belief space. Similarly, we compute a valid policy over a goal-constrained belief space, which contains beliefs visited by desired executions that can achieve the safe-reachability objective. The goal-constrained belief space is generally much smaller than the original belief space.

Our synthesis algorithm, bounded policy synthesis (BPS), computes a valid policy by iteratively searching for a candidate plan in the goal-constrained belief space and constructing a policy from this candidate plan. We compactly represent the goal-constrained belief space over a bounded horizon using symbolic constraints. The applicability of constraint-based methods has been already advocated in several robotics planning algorithms [8, 23, 31, 44]. Many of these algorithms take advantage of a modern, incremental SMT solver [9] for efficiency. Inspired by this, we apply the SMT solver to efficiently explore the symbolic goal-constrained belief space to generate candidate plans. Note that a candidate plan is a single path that only covers a particular observation at each step, while a valid policy is contingent on all possible observations. Therefore, once a candidate plan is found, BPS tries to generate a valid policy from the candidate plan by considering all possible events at each step. If this policy generation fails, BPS adds additional constraints that block invalid plans and force the SMT solver to generate other better plans. The incremental capability of the SMT solver allows BPS to efficiently generate alternate candidate plans when we update the constraints. If there is no new candidate plan for the current horizon, BPS increases the horizon and repeats the above steps until it finds a valid policy or reaches a given horizon bound.

In summary, the contributions of the paper are:

- We show that in some domains, our formulation of POMDPs with safe-reachability objectives offers a better guarantee of both safety and reachability than the existing POMDP models through an example (Section 4).
- We introduce the notion of a goal-constrained belief space to address the scalability challenge of solving POMDPs with safe-reachability objectives. Based on this notion, we present a novel approach called BPS for policy synthesis of POMDPs with safe-reachability objectives.
- We evaluate the scalability of BPS using a case study involving a partially observable robotic domain with uncertain obstacles (Figure 1). The experimental results demonstrate that BPS can scale up to huge belief spaces by focusing on the goal-constrained belief space.

2 RELATED WORK

POMDPs [38] provide a principled mathematical framework for modeling a variety of robotics problems in the face of uncertainty. Many POMDP algorithms [1, 15, 25, 26, 39, 40] for robot applications focus on discounted reward objectives. Recent work [5, 6, 42] has investigated almost-sure satisfaction of POMDPs with temporal logic specifications, where the goal is to check whether a temporal logic objective can be ensured with probability 1. Our approach can be seen as synthesizing policies for large POMDP problems with basic temporal logic objectives (safe-reachability), but not limited to almost-sure satisfaction analysis. Though we may formulate a safe-reachability objective as an optimization problem by assigning negative rewards for unsafe states and positive rewards for goal states, this formulation does not always yield good policies [43].

Recently, there has been a large body of work that extends the traditional POMDP model with notions of risk and cost, including constrained POMDPs (C-POMDPs) [21, 24, 36, 43], risk-sensitive POMDPs (RS-POMDPs) [20, 28] and chance-constrained POMDPs (CC-POMDPs) [37]. There are two major differences between their models and our formulation of POMDPs with safe-reachability objectives. First, the objective of these models is to maximize the cumulative expected reward while keeping the expected cost/risk below some threshold, while in our case, the objective is to satisfy a safe-reachability objective in all possible executions including the worst case, providing a better safety guarantee than the formulation of expected cost/risk threshold constraints. Second, C/RS/CC-POMDPs typically need to assign a proper positive reward for goal states to ensure reachability and do not have direct control over the probability of reaching goal states (e.g., reach a goal state with a probability greater than some threshold), while our safe-reachability objective can directly encode this probability threshold.
constraint as a boolean requirement, providing a better reachability
guarantee than the quantitative formulation of C/RS/CC-POMDPs. While C/RS/CC-POMDPs are suitable for many applications, there are domains in robotics such as autonomous driving and disaster rescue that demand synthesis of policies that can provide such strong guarantee of reaching goal states safely.

Task and Motion Planning (TMP) [2, 8, 13, 14, 18, 19, 23, 41, 44] describes a class of challenging problems that combine low-level motion planning and high-level task reasoning. Most of these TMP approaches focus on deterministic domains, while several of them apply to uncertain domains with uncertainty in perception [18, 23]. The main difference is that, the above works perform online planning with a determinized approximation of belief space dynamics [35] assuming the most likely observation will be obtained, while our approach synthesizes a valid policy offline contingent on all possible events.

Our method computes a valid policy by iteratively searching for a candidate plan that is likely to succeed with determined observations in the goal-constrained belief space, and then constructing a policy from this candidate plan by considering other possible observations. This idea has been shown to improve the scalability of algorithms for a variety of uncertain domains [4, 10, 30]. The scalability of our approach also relies on exploiting the notion of a goal-constrained belief space. This idea resembles efficient point-based POMDP algorithms [25, 26] based on (optimally) reachable belief space.

We apply techniques from Bounded Model Checking (BMC) [3] to compactly represent the goal-constrained belief space over a bounded horizon. BMC verifies whether a finite state system satisfies a given temporal logic specification. Thanks to the tremendous increase in the reasoning power of practical SMT (SAT) solvers, BMC can scale up to large systems with hundreds of thousands of states. Our approach efficiently explores the goal-constrained belief space by leveraging a modern, incremental SMT solver [9]. It of states. Our approach efficiently explores the goal-constrained belief space over a compactly represent the goal-constrained belief space in belief space by leveraging a modern, incremental SMT solver [9]. It assuming the most likely observation will be obtained, while our approach synthesizes a valid policy offline contingent on all possible events.

Our method computes a valid policy by iteratively searching for a candidate plan that is likely to succeed with determined observations in the goal-constrained belief space, and then constructing a policy from this candidate plan by considering other possible observations. This idea has been shown to improve the scalability of algorithms for a variety of uncertain domains [4, 10, 30]. The scalability of our approach also relies on exploiting the notion of a goal-constrained belief space. This idea resembles efficient point-based POMDP algorithms [25, 26] based on (optimally) reachable belief space.

3 PROBLEM FORMULATION

In this work, we consider the problem of policy synthesis for POMDPs:

Definition 3.1 (POMDP).
A Partially Observable Markov Decision Process (POMDP) is a tuple $P = (S, A, T, O, Z)$:

- $S$ is a finite set of states.
- $A$ is a finite set of actions.
- $T$ is a probabilistic transition function $T(s, a, s') = p(s'|s, a)$, which defines the probability of moving to state $s'$ in $S$ after taking an action $a \in A$ in state $s \in S$.
- $O$ is a finite set of observations.
- $Z$ is the probabilistic observation function $Z(s', a, o) = p(o|s', a)$, which defines the probability of observing $o \in O$ after taking an action $a \in A$ and reaching state $s' \in S$.

Due to uncertainty in transition and observation, the actual state is partially observable and typically we maintain a belief, which is a probability distribution over all possible states $b : S \rightarrow [0, 1]$ with $\sum_{s \in S} b(s) = 1$. The set of beliefs $B = \{b : S \rightarrow [0, 1] | \sum_{s \in S} b(s) = 1\}$ is known as belief space. Note that a transition $T_b$ in belief space is a deterministic function $b' = T_b(b, a, o)$, i.e., given an action $a \in A$ and an observation $o \in O$, the updates to beliefs are deterministic based on the formula:

$$b'(s') = \alpha Z(s', a, o) \sum_{s \in S} T(s, a, s') b(s)$$

where $\alpha$ is a normalization constant.

Definition 3.2 (Plan).
A plan in belief space is a sequence $\sigma = (b_0, a_1, o_1, b_1, a_2, o_2, b_2, \ldots)$ such that for all $i > 0$, the belief updates satisfy the transition function $T_b$, i.e., $b_i = T_b(b_{i-1}, a_i, o_i)$, where $a_i \in A$ is an action and $o_i \in O$ is an observation.

Definition 3.3 (Policy).
A policy $\pi : B \rightarrow A$ is a function that maps a belief $b \in B$ to an action $a \in A$. A policy $\pi$ defines a set of plans in belief space: $\Omega_\pi = \{ \sigma = (b_0, a_1, o_1, \ldots) | \forall i > 0, a_i = \pi(b_{i-1}) \text{ and } o_i \in O \}$. For each plan $\sigma \in \Omega_\pi$, the action $a_i$ at each step $i$ is chosen by the policy $\pi$.

3.1 Safe-Reachability Objective

In this work, we consider POMDPs with safe-reachability objectives:

Definition 3.4 (Safe-Reachability Objective).
A safe-reachability objective is a tuple $G = (Dest, Safe)$:

- Safe is a set of safe beliefs
- Dest is a set of goal beliefs.

A safe-reachability objective $G$ compactly represents the set $\Omega_G$ of satisfiable plans in belief space:

Definition 3.5 (Satisfiable Plan).
A plan $\sigma = (b_0, a_1, o_1, \ldots)$ satisfies a safe-reachability objective $G = (Dest, Safe)$ if there exists a belief $b_k$ at step $k$ in the plan $\sigma$ that is a goal belief $b_k \in Dest$ and all the beliefs $b_i (i < k)$ visited before step $k$ are safe beliefs $b_i \in Safe$.

Note that safe-reachability objectives are defined using sets of beliefs (probability distributions). The qualitative analysis problem of POMDPs with requirements of a goal state is eventually reached with a probability above some threshold while keeping the probability of visiting unsafe states below some threshold, can be easily formulated as a safe-reachability objective $G = (Dest, Safe)$ defined as follows:

$$Dest = \{ b \in B | \sum_{s \text{ is a goal state}} b(s) > 1 - \delta_1 \}$$

$$Safe = \{ b \in B | \sum_{s \text{ violates safety}} b(s) < \delta_2 \}$$

Where $\delta_1$ and $\delta_2$ are small values that represents tolerance.
we can construct a goal-constrained belief space. Thus, computing policies over the goal-constrained belief space is a function that maps a belief $b \in B$ in state $\Omega$ to an action $a \in A$. For each edge, the first line is the transition probability and the second line is the tuple of observation probabilities $(p_{pos}, p_{neg})$.

### 3.2 Goal-Constrained Belief Space

It is intractable to compute a full policy that satisfies a given safe-reachability objective for POMDPs, even under the assumption of bounded horizon, due to the curse of dimensionality [32]: the belief space $B$ is a high-dimensional, continuous space that contains an infinite number of beliefs.

However, the reachable belief space [25] $\mathcal{R}(b_{init})$ that contains beliefs reachable from the given initial belief $b_{init}$, is much smaller than $B$ in general. Moreover, the safe-reachability objective $G$ defines a set $\Omega_G$ of plans that satisfy $G$. Combining $\mathcal{R}(b_{init})$ and $\Omega_G$, we can construct a goal-constrained belief space $\mathcal{R}^*(b_{init}, G)$ that contains beliefs reachable from the initial belief $b_{init}$ under satisfiable plans $\sigma \in \Omega_G$. The goal-constrained belief space $\mathcal{R}^*(b_{init}, G)$ is usually much smaller than the reachable belief space $\mathcal{R}(b_{init})$. Thus, computing policies over the goal-constrained belief space $\mathcal{R}^*(b_{init}, G)$ can lead to a substantial gain in efficiency.

### 3.3 Problem Statement

Given a POMDP $P = (S, A, T, O, Z)$, an initial belief $b_{init}$ and a safe-reachability objective $G$, our goal is to synthesize a valid policy $\pi_R^*$ over the corresponding goal-constrained belief space $\mathcal{R}^*(b_{init}, G)$:

**Definition 3.6 (Valid Policy).** A valid policy $\pi_R^* : \mathcal{R}^*(b_{init}, G) \rightarrow A$ over a goal-constrained belief space is a function that maps a belief $b \in \mathcal{R}^*(b_{init}, G)$ to an action $a \in A$. Therefore, the set $\Omega_{\pi_R^*}$ of plans defined by the policy $\pi_R^*$ is a subset of the set $\Omega_G$ defined by the safe-reachability objective $G$, i.e., $\Omega_{\pi_R^*} \subseteq \Omega_G$.

### 4 Related to Unconstrained POMDPs

There are two distinct approaches that can model safe-reachability objectives *implicitly* using the existing POMDP models in the literature. The first approach is to incorporate safety and reachability constraints as negative penalties for unsafe states and positive rewards for goal states in *unconstrained* POMDPs with quantitative objectives. However, the authors of [43] have shown a counterexample that demonstrates formulating constraints as unconstrained POMDPs with quantitative objectives does not always yield good policies. The second approach is to encode safe-reachability objectives implicitly as C/RS/CC-POMDPs that extend unconstrained POMDPs with notions of risk and cost [20, 21, 24, 28, 36, 37, 43]. In this section, we show the differences between POMDPs with safe-reachability objectives and unconstrained/C/RS/CC-POMDPs through an example.

In Figure 1, after the robot passes the yellow “shadow” region and moves to the position where it is ready to pick up a green cup from the black storage area (start state $s_{\text{ready}}$), it needs to decide how to pick up the object. There are two action choices: picking up the left hand (action $a_L$) and picking up the right hand (action $a_R$). Both $a_L$ and $a_R$ are uncertain, and the robot may hit the storage while executing $a_L$ or $a_R$, which results in an unsafe collision state $s_{\text{unsafe}}$. There are two possible observations after executing $a_L$ or $a_R$: observation $o_{pos}$ representing the robot observes a cup in its hand and observation $o_{neg}$ representing the robot observes no cup in its hand (Note that the actual state may be different from the observation due to uncertainty). The task objective is to reach a goal state $s_{\text{goal}}$ where the robot holds a cup in its hand with a probability greater than 0.8 (reachability) while...
keeping the probability of visiting unsafe state $s_{\text{unsafe}}$ below the threshold 0.2 (safety). The probability transition and observation functions are shown in Figure 2. Based on Formula 1, we can get the transition in the corresponding belief space (see Figure 3).

If we model this problem as an unconstrained POMDP by assigning a negative penalty $-P (P > 0)$ for unsafe state $s_{\text{unsafe}}$ and a positive reward $R (R > 0)$ for goal state $s_{\text{goal}}$, the optimal action for $s_{\text{ready}}$ that achieves the maximum reward is always $a_L$, no matter what values of $P$ and $R$ are. This is because the expected reward of action $a_L (0.9R - 0.1P)$ is greater than the expected reward of $a_R (0.85R - 0.1P)$. However, action $a_L$ does not satisfy the original safe-reachability objective in the worst case where the robot observing $o_{\text{neg}}$ after executing action $a_L$ and the resulting belief state $(0, 0.28, 0.72)$ violates the original safety-reachability objective.

If we model this problem as a C/RS/CC-POMDP by assigning a positive reward $R$ for goal state $s_{\text{goal}}$ and a cost $1$ for visiting unsafe state $s_{\text{unsafe}}$, the best action for $s_{\text{ready}}$ will be $a_L$ since both $a_L$ and $a_R$ satisfies the cost/risk constraint (expected cost/risk $0.1 < 0.2$) and the expected reward of $a_L (0.9R)$ is greater than the expected reward of $a_R (0.85R)$. However, action $a_L$ violates the original safe-reachability objective for the same reason explained above.

On the other hand, using our formulation of POMDPs with safe-reachability objectives, the best action for $s_{\text{ready}}$ will be $a_R$. This is because, as shown in Definition 3.6, a valid policy in our formulation should satisfy the safe-reachability objective in all possible executions and only $a_R$ satisfies the safe-reachability objective in every possible execution.

The intent of this simple example is to illustrate that in some domains where we want the robot to safely accomplish the task, our formulation of POMDPs with safe-reachability objectives can provide a better guarantee of both safety and reachability than the existing POMDP models. While the formulations of cost/risk as negative penalties in unconstrained POMDPs and expected cost/risk threshold constraints in C/RS/CC-POMDPs are suitable for many applications, there are domains such as autonomous driving and disaster rescue that demand synthesis of policies that can provide such strong guarantee of reaching goal states safely as in our formulation, especially when violating safety requirements results in irreversible damage to robots.

5 BOUNDED POLICY SYNTHESIS

The core steps of BPS (Algorithm 1) are shown in Figure 4. BPS computes a valid policy by iteratively searching for a candidate plan in the goal-constrained belief space $\mathcal{R}^*(b_{\text{init}}, \mathcal{G})$ and constructing a valid policy from this candidate plan. Figure 5 graphically depicts one example run of BPS.

First BPS compactly encodes the goal-constrained belief space $\mathcal{R}^*(b_{\text{init}}, \mathcal{G})$ (the black box in Figure 5) w.r.t. the given POMDP $P = (S, A, T, O, Z)$, the initial belief $b_{\text{init}}$ and the safe-reachability objective $\mathcal{G}$ over a bounded horizon $k$ as a logical formula $\Phi_k$ (Algorithm 1, lines 2, 6, 8). We describe the details of the constraints that encode the goal-constrained belief space in Section 5.1.

Then BPS computes a candidate plan by checking the satisfiability of the constraint $\Phi_k$ (line 10) through a modern, incremental SMT solver [9]. Note that the horizon $k$ restricts the length of the plan and thus the robot can only execute $k$ actions.

If $\Phi_k$ is satisfiable, the SMT solver returns a candidate plan (the dashed green path in Figure 5) and BPS tries to generate a valid policy from the candidate plan by considering all possible observations, i.e., other branches following the red observation node at each step (line 14). If this policy generation succeeds, we find a valid policy. Otherwise, BPS adds additional constraints that block this invalid plan (line 16) and forces the SMT solver to generate another better candidate.

If $\Phi_k$ is unsatisfiable and thus there is no new plan for the current horizon, BPS increases the horizon by one (line 21) and repeats the above steps until a valid policy is found (line 18) or a given horizon bound $h$ is reached (line 3).

This incremental SMT solver [9] can efficiently generate alternate candidate plans by maintaining a stack of scopes, where each scope is a container for a set of constraints and the corresponding "knowledge" learned from this set of constraints. For fast repeated satisfiability checks, when we update constraints (lines 2, 6, 8, 16), rather than rebuilding the "knowledge" from scratch, the incremental SMT solver only changes the "knowledge" related to the updates by pushing (line 7) and popping (line 20) scopes. Thus the "knowledge" learned from previous satisfiability checks can be reused.
5.1 Constraint Generation

In the first step, we use an encoding from Bounded Model Checking (BMC) [3] to construct the constraint $\Phi_k$ representing the goal-constrained belief space $R^*(b_{init}, G)$ w.r.t. the POMDP $P = (S, A, T, O, Z)$, the initial belief $b_{init}$ and the safe-reachability objective $G$ over the bounded horizon $k$. The idea behind BMC is to find a finite plan with increasing horizon that satisfies the given safe-reachability objective.

The constraint $\Phi_k$ contains three parts:

1. Starting from the initial belief (line 2): $b_k = b_{init}$.
2. Unfolding of the transition up to the horizon $k$ (line 6):
   $$ \bigwedge_{i=1}^k (b_i = T_R(b_{i-1}, a_i, o_i)) $$
3. Satisfying the safe-reachability objective $G$ (line 8).

We can translate a safe-reachability objective to the constraint $G(\sigma_k, G, k)$ on bounded plans $\sigma_k = (b_s, a_{s+1}, o_{s+1}, \ldots, a_k, o_k, b_k)$ using the rules provided by BMC [3] as follows:

$$ G(\sigma_k, G, k) = \bigwedge_{i=s}^{i-1} (b_i = Dest \land (\bigvee_{j=s}^{i-1} (b_j \in Safe))) $$  

For a safe-reachability objective $G$ with a set $Dest$ of goal beliefs and a set $Safe$ of safe beliefs, a finite plan that visits a goal belief while staying in the safe region is sufficient to satisfy $G$. Therefore, we only need to specify that a bounded plan with length $k$ eventually visits a belief $b_i \in Dest$ while staying in the safe region $\bigwedge_{j=s}^{i-1} (b_j \in Safe)$, as shown in Formula 4.

5.2 Plan Generation

The next step is to generate a candidate plan $\sigma_k$ of length $k$ that satisfies the constraint $\Phi_k$. We apply an incremental SMT solver to efficiently search for such a candidate in the goal-constrained belief space $R^*(b_{init}, G)$ defined by $\Phi_k$ (line 10). If $\Phi_k$ is unsatisfiable, there is no bounded plan $\sigma_k$ for the current horizon. In this case, we need to increase the horizon (line 21). If $\Phi_k$ is satisfiable, the SMT solver will return a satisfying model that assigns concrete values $b_i^{o_k}, a_{i+1}^{o_k}$ and $a_{i+1}^{o_k}$ for the belief $b_i$, action $a_{i+1}$ and observation $o_{i+1}$ at each step $i$ respectively, which can be used to construct the candidate plan $\sigma_k = (b_s^{o_k}, a_{s+1}^{o_k}, o_{s+1}^{o_k}, a_{s+1}^{o_k}, \ldots, a_k^{o_k}, o_k^{o_k}, b_k^{o_k})$.

5.3 Policy Generation

After plan generation, we get a candidate plan $\sigma_k$ (the dashed green path in Figure 5) that satisfies the safe-reachability objective $G$. This candidate plan is a single path that only covers a particular observation $o_i^{o_k}$ at each step $i$. To construct a valid policy, we should also consider other possible observations $o'_i \neq o_i^{o_k}$, i.e.,

\begin{algorithm}[h]
\caption{Algorithm 1: BPS}
\begin{algorithmic}[1]
\Require
POMDP $P = (S, A, T, O, Z)$
Initial Belief $b_{init}$
Safe-Reachability Objective $G$
\Ensure A Valid Policy $\pi$
\State $k \leftarrow s$; /* Initial horizon */
\State $\Phi_k \leftarrow (b_s = b_{init})$; /* Initial belief */
\While {$k \leq h$}
\State $\sigma_k \leftarrow \emptyset$; /* $\sigma_k$: Candidate plan */
\State if $k > s$ then
\State \Comment{Add transition at step $k$ if $k > s$} /*
\State $\Phi_k \leftarrow \Phi_k \land (b_k = T_R(b_{k-1}, a_k, o_k))$;
\State \Comment{Push scope} /*
\State $\Phi_k \leftarrow \Phi_k \land \sigma_k(G, k)$;
\State $\emptyset \leftarrow \sigma_k$ /* Candidate generation */
\State $\sigma_k \leftarrow \text{IncrementalSMT}(\Phi_k)$;
\State if $\emptyset = \sigma_k$ then /* No new plan */
\State \Comment{Break} /*
\State break;
\Else /* $\emptyset$: constraints for blocking invalid plans */
\State $\pi, \phi = \text{PolicyGeneration}(P, G, \sigma_k, s + 1, k)$;
\State if $\emptyset \neq \phi$ then /* Generation failed */
\State $\Phi_k \leftarrow \Phi_k \land \phi$;
\State else
\State \Comment{Return} /*
\State $\return \pi$;
\State $\sigma_k \leftarrow \emptyset$;
\State \Comment{Pop scope: pop goal and $\phi$ at step $k$} /*
\State pop($\Phi_k$);
\State $k \leftarrow k + 1$; /* Increase horizon */
\EndWhile
\State $\return \emptyset$;
\end{algorithmic}
\end{algorithm}

\begin{algorithm}[h]
\caption{Algorithm 2: PolicyGeneration}
\begin{algorithmic}[1]
\Require
POMDP $P = (S, A, T, O, Z)$
Safe-Reachability Objective $G$
Candidate Plan $\sigma_k = (b_s^{o_k}, a_{s+1}^{o_k}, o_{s+1}^{o_k}, a_{s+1}^{o_k}, b_{k+1}^{o_k} \ldots)$
\Ensure A Valid Policy $\pi$ and Constraints $\phi$ for blocking invalid plans if the input candidate plan is invalid
\State $\pi \leftarrow \emptyset$;
\For{$i = h$ downto $s$}
\State \Comment{Try observation $o$} /*
\State $b_i^{o_k} \leftarrow T_R(b_{i-1}, a_i^{o_k}, o_i)$;
\State \Comment{Call BPS to construct the branch} /*
\State $\pi' \leftarrow \text{BPS}(P, b_i^{o_k}, G, i, h)$;
\State if $\emptyset = \pi'$ then /* Construction failed */
\State \Comment{Construct $\phi$ using Formula 5} /*
\State Construct $\phi$ using Formula 5
\State \Comment{Return} /*
\State $\return \emptyset, \phi$;
\EndFor
\Comment{Combine policy} /*
\State $\pi \leftarrow \pi \cup \pi'$;
\Comment{Record action choice for belief $b_i^{o_k}$} /*
\State $\pi(b_i^{o_k}) \leftarrow a_i^{o_k}$;
\State $\return \pi, \emptyset$;
\end{algorithmic}
\end{algorithm}
other branches following the red observation node for each step i. Policy generation (Algorithm 2) tries to construct a valid policy from a candidate plan by considering all possible observations at each step.

For a candidate plan $\sigma_k$, we process each step of $\sigma_k$, starting from the last step (Algorithm 2, line 24). For each step $i$, since the set of observations $O$ is finite, we can enumerate every possible observation $o_i^t \neq o_i^m$ (line 25) and compute the next belief $b_i^t$ using the transition function (line 26). To ensure the action $a_i^m$ also works for this different observation $o_i^t$, we need to compute a valid policy for the branch starting from $b_i^t$, which is another BPS problem and can be solved using Algorithm 1 (line 27).

If we successfully construct the valid policy $\pi'$ for this branch, we can add $\pi'$ to the policy $\pi$ for the original synthesis problem (line 31). Otherwise, this candidate plan is invalid for the original synthesis problem (line 31). Otherwise, this candidate plan $\sigma_k$ can not be an element of a valid policy $\sigma_k \notin \Omega$. In this case, we know that the prefix of the candidate plan $(b_i^t, a_i^m, b_{i+1}^t, \ldots, b_{k-1}^t, a_k^m)$ is invalid for current horizon $k$ and we can add additional constraints $\phi$ to block all invalid plans that have this prefix (line 29):

$$\phi = \neg(b_i^t \land \langle a_i = a_i^m \land \bigwedge_{m=r+1}^{i-1} (a_m = a_m^o \land o_m = o_m^o) \land (b_m = b_m^o) \rangle)$$

(5)

Note that $\phi$ is only valid for current horizon $k$ and when we increase the horizon, we should pop the scope related to the additional constraints $\phi$ from the stack of the SMT solver (line 20) so that we can revisit this prefix with the increased horizon. If we successfully construct policies for all other branches at step $i$, we know that the choice of action $a_i^m$ for belief $b_i^t$ is valid for all possible observations. Then we record this choice for belief $b_i^t$ in the policy (line 32). This policy generation terminates when it reaches the start step $s$ as stated in the for-loop (line 24) or it fails to construct the valid policy $\pi'$ for a branch (line 28).

### 5.4 Algorithm Complexity

The reachable belief space $R(b_{\text{init}})$ can be seen as a tree where the root node is the initial belief $b_{\text{init}}$ and at each node, the tree branches on every action and observation. The given horizon bound $h$ limits the height of the tree. Therefore, the reachable belief space $R_k(b_{\text{init}})$ of height $h$ contains $O(|A|^h|O|^h)$ plans, where $|A|$ and $|O|$ are the size of action set $A$ and the size of observation set $O$ respectively. To synthesize a valid policy, a naive approach that checks every plan in the reachable belief space $R_k(b_{\text{init}})$ requires $O(|A|^h|O|^h)$ calls to the SMT solver. This exponential growth of the reachable belief space $R_k(b_{\text{init}})$ due to branches on both action and observations is a major challenge for synthesizing a valid policy.

In our case, BPS exploits the notion of goal-constrained belief space $R_k^G(b_{\text{init}}, G)$ and efficiently explores the goal-constrained belief space $R_k^G(b_{\text{init}}, G)$ by leveraging an incremental SMT solver to generate a candidate plan $\sigma$ of length at most $h$. This candidate plan fixes the choice of actions at each step and thus the policy generation process only needs to consider the branches on observations for each step, as shown in Figure 5. Therefore, BPS requires $O(|O|^h)$ calls to the SMT solver, where $I$ is the number of interactions between plan generation and policy generation, while the naive approach described above requires $O(|A|^h|O|^h)$ SMT solver calls. In general, $I$ is often much smaller than $|A|^h$, which leads to much faster policy synthesis. Therefore, we expect our method to be effective for POMDPs with a high-dimensional action space and a restricted partially observable component, but would not scale well for POMDPs with high-dimensional/continuous observation space.

### 6 EXPERIMENTS

We evaluate BPS in a partially observable kitchen domain (Figure 1) with a PR2 robot and $M$ uncertain obstacles placed in the yellow "shadow" region. The task for the robot is to safely pass the yellow "shadow" region avoiding collisions with uncertain obstacles and eventually pick up a green cup from the black storage area.

We first discretize the kitchen environment into $N$ cells. We assume that the locations of the obstacles are uniformly distributed among the cells in the yellow "shadow" region and there is at most one obstacle in each cell. We also assume the robot starts at a known initial location. However, due to the robot's imperfect perception, the locations of the robot, the locations of uncertain obstacles, and the location of the target cups are all partially observable during execution.

In this domain, the actuation and perception of the robot are imperfect. There are ten uncertain robot actions ($|A| = 10$):

1. Four move actions that move the robot to an adjacent cell in four directions: including move-north, move-south, move-west and move-east. Move actions could fail with a probability $p_{\text{fail}}$, resulting in no change in the state.
2. Four look actions that observe a cell to see whether there is an obstacle in that cell, including look-north, look-south, look-west, look-east (look at the adjacent cell in the corresponding direction). When the robot calls look to observe a particular cell, it may either make an observation $o = o_{\text{pos}}$ representing the robot observes an obstacle in cell $i$, or $o = o_{\text{neg}}$ representing the robot observers no obstacle in cell $i$. The probabilistic observation function $Z(s, a, o)$ for look actions is defined based on the false positive probability $p_{\text{pos}}$ and the false negative probability $p_{\text{neg}}$.
3. Two pick-up actions that pick up an object from the black storage area: pick-up using the left hand $a_L$ and pick-up using the right hand $a_R$. The model of pick-up actions is the same as what we discussed in Section 4 (see Figure 2).

The task shown in Figure 1 can be specified as a safe-reachability objective with a set $\text{Dest}$ of goal beliefs and a set $\text{Safe}$ of safe beliefs, defined as follows:

$$\text{Dest} = \{ b \in B \mid \sum b(\text{target cup in robot’s hand}) > 1 - \delta_1 \}$$

$$\text{Safe} = \{ b \in B \mid \sum b(\text{robot in collision}) < \delta_2 \}$$

(6)

where $\delta_1$ and $\delta_2$ are small values that represent tolerance. The reachability objective specifies that in a goal belief, the probability of the robot picking up the target cup should be greater than the threshold $1 - \delta_1$. The safety objective specifies that in a safe belief, the probability of the robot in collision (the robot and one obstacle in the same cell) should be less than the tolerance $\delta_2$. 
We present a novel policy synthesis method called BPS for POMDPs with safe-reachability objectives. We exploit the notion of a goal-constrained belief space to improve computational efficiency. We construct constraints in a way similar to Bounded Model Checking theories solver [9]. We evaluate BPS in an uncertain robotic domain and the results show that our method can synthesize policies for large problems by focusing on the goal-constrained belief space.

The current implementation of BPS operates on an exact representation of the policy (the tree structure shown in Figure 5). As a result, BPS suffers from the exponential growth as the horizon increases. An important ongoing question is how to approximately represent the policy while preserving correctness. Another issue arises from the discrete representations (discrete POMDPs) used in our approach. While many robot tasks can be modeled using these representations, discretization often suffers from the "curse of dimensionality". Investigating how to deal with continuous state spaces and continuous observations directly without discretization is another promising future direction for this work and its application in robotics.

8 ACKNOWLEDGMENTS
This work was supported in part by NSF CCF 1139011, NSF CCF 1514372, NSF CCF 1162076 and NSF IIS 1317649.