To appear in the proceedings of the 2014 Workshop on the Algorithmic Foundations of Robotics (WAFR 2014)

Asymptotically Optimal Stochastic Motion Planning with Temporal Goals

Ryan Luna, Morteza Lahijanian, Mark Moll, and Lydia E. Kavraki

Department of Computer Science, Rice University
{rluna,morteza,mmoll,kavraki}@rice.edu

Abstract. This work presents a planning framework that allows a robot with stochastic action uncertainty to achieve a high-level task given in the form of a temporal logic formula. The objective is to quickly compute a feedback control policy to satisfy the task specification with maximum probability. A top-down framework is proposed that abstracts the motion of a continuous stochastic system to a discrete, bounded-parameter Markov decision process (BMDP), and then computes a control policy over the product of the BMDP abstraction and a DFA representing the temporal logic specification. Analysis of the framework reveals that as the resolution of the BMDP abstraction becomes finer, the policy obtained converges to optimal. Simulations show that high-quality policies to satisfy complex temporal logic specifications can be obtained in seconds, orders of magnitude faster than existing methods.

Keywords: planning under uncertainty, temporal logic planning, stochastic systems, formal control synthesis

1 Introduction

Robots are rapidly becoming capable of performing a wide range of tasks with a high-degree of autonomy. There is a growing desire to take full advantage of these systems by allowing a human operator to dictate a high-level task to the robot and let the robot itself decide the low-level details of how to accomplish the task. Consider an automated warehouse where items are retrieved by a robot and then dropped off at a central location for further processing. A single human dispatcher can coordinate such tasks at a high-level by simply telling the robot *which* items to gather. This is in contrast to lower-level coordination where a technically savvy or highly trained operator must tell the robot *how* to gather each item. By abstracting the motion planning objective into a high-level task, the need for a human operator to reason over low-level details (e.g., the order items are gathered) is obviated. There are two fundamental challenges, however, that inhibit this high-level abstraction. First, translating a high-level specification into an equivalent model fit for a motion planning algorithm is a computationally difficult endeavor, typically an exponential-time operation [1]. Second, physical robots suffer from uncertainties that can invalidate a motion plan, like noisy actuation, unreliable sensing, or a changing environment, and robustly handling uncertainty can require significant computation time [2]. Extensive literature exists for solving these challenges in isolation, but methods that are both efficient and effective at high-level task planning for an uncertain system remain elusive.

High-level specifications using temporal logics have been employed to improve the expressiveness of a motion planning task (e.g., [3–10]). These logics allow for a natural encoding of both Boolean and temporal constraints, and the classic motion planning task of *move from start to goal without collision* can be greatly enhanced using these operators. For instance, in the warehouse scenario described above, complex tasks such as

> *"Pick up items from locations A, B, and C, in any order, and drop them off at location D"* or
> *"Pick up items from locations A or B and then C and drop them off in D; meanwhile, if B is ever visited, then avoid E"*

are easily encoded using only temporal and Boolean operators. Given a motion planning specification in the form of a temporal logic formula, existing frameworks (e.g., [3–10]) consider a mixed discrete and continuous approach, where Boolean propositions are mapped to discrete regions of the state space and planning is performed in the continuous space to satisfy the specification.

When the robot is subject to action uncertainty, robust motion planners have been developed that compute a control strategy over the entire state space rather than a single trajectory (e.g., [11–13]). This strategy is often referred to as a *policy*. Conceptually, a policy is a lookup table that maps each state to a particular action. An *optimal policy* maximizes the *reward* the robot can expect to receive given a stochastic motion model of its evolution. Computing an optimal policy can take significant time, however, because every state of the system must be reasoned over to ensure the action selected is indeed optimal.

This work operates at the intersection of high-level task planning and planning under action uncertainty. A top-down framework is presented that is capable of quickly computing an optimal control policy that satisfies a temporal logic specification with maximum probability by utilizing a combination of discrete and continuous space planning. To robustly handle noise in the actuation of the robot, the method constructs an abstraction in the form of an *uncertain Markov model* that models the evolution of the robot as it moves between discrete regions of the state space. Given a temporal logic task specification, the framework then constructs an equivalent *deterministic finite automaton* (DFA) that expresses the task and computes an optimal control policy over the product of the DFA and the discrete abstraction to maximize the probability of satisfying the specification.

1.1 Related Work

Motion planning for realistic robotic tasks is the subject of a large body of recent work known as *formal methods in robotics* [3–10]. The kinds of tasks that are studied typically admit a wide latitude of possible solutions; this is evident in the tasks described earlier for the warehouse scenario. Many complex motion planning scenarios can be naturally translated to temporal logics, in particular *linear temporal logic* (LTL) [14]. Unfortunately, temporal logic planning suffers from *state space explosion*, and existing methods rely on a discrete abstraction of the continuous system to gain computational tractability.

One class of methods for temporal logic planning synthesize controllers over a discrete abstraction of the state space [3, 5]. The relationship between the controllers and a discretization of the space ensures that motion between adjacent regions is realizable by the continuous system, known as a *bisimilar* abstraction. Synthesis of *reactive* controllers have also been considered that allow for robust control in a dynamic environment, provided that all environmental behaviors are also encoded in temporal logic [4, 6, 15, 16]. These methods are *correct-by-construction*, and find a satisfying trajectory if one exists. Synthesizing controllers that satisfy the bisimilarity constraints, however, admits only simple dynamical models. Recent work attempts reactive synthesis for non-linear systems [17], but constructing these controllers remains computationally difficult.

Sampling-based motion planners have been augmented to satisfy a task specification given in LTL [7, 8, 10, 18, 19]. These works are able to quickly emit a satisfying trajectory for systems with hybrid and/or complex dynamics. Note that these methods are not correct-by-construction. The probabilistic completeness of many sampling-based planners, however, guarantees that if a satisfying trajectory exists, the probability of finding a trajectory grows to 1 over time.

The temporal logic planning methods described above do not address instances where the robot suffers from uncertainties. When there is uncertainty in actuation, methods exist for temporal logic planning that employ a Markov decision process (MDP) to model the evolution of the system through the state space [20, 21]. The goal in these methods is to compute a control policy over the MDP abstraction to satisfy a high-level task with maximum probability. These works are incomplete, however, in that methods to construct the approximating MDP for the robot are not presented; only planning over an existing abstraction is discussed. *Uncertain* MDPs, where transition probabilities can belong to sets of values, have also been employed to provide a hierarchical abstraction and improve computational complexity [22]. Strong assumptions must be made on the structure of this abstraction, many of which are difficult to realize for physical systems.

Construction of a Markov abstraction for continuous-time and space systems has been studied in the literature for stochastic optimal control. In the *stochastic motion roadmap* (SMR) [11], the state space is discretized through sampling and an MDP is constructed over the sampled states using a Monte Carlo simulation; a set of discrete actions is assumed. Another method is the *incremental* MDP (iMDP) algorithm [12], which asymptotically approximates the optimal policy of the continuous stochastic system by sampling both a state and a set of candidate controls; a single control is chosen for the state with *value iteration*. To ensure a good approximation of the optimal policy, both SMR and iMDP construct a highly accurate MDP abstraction. Achieving the Markov property exactly, however, requires very dense state space sampling. Recent work suggests the use of a *bounded-parameter* Markov decision process (BMDP) [23], a special class of uncertain MDPs which can be solved in polynomial-time with respect to the number of states, as the abstraction model [13, 24]. A BMDP allows for coarse discretization of the state space by relaxing the Markov constraint while still fully representing the memoryless transition model. Moreover, a BMDP does not have the strong assumptions on the transition model that general uncertain MDPs do.

1.2 Contribution

This paper introduces a planning framework that quickly computes a control policy for a system with uncertain actuation to satisfy a high-level specification with maximum probability. The proposed planning framework utilizes a coarse Markov abstraction to mitigate state space explosion when planning for the continuous stochastic system. Unlike previous works in temporal logic planning, however, the proposed framework makes few assumptions on the underlying dynamics, and is applicable to a broad class of stochastic systems. The proposed method builds upon previous work [13, 24] by constructing a coarse, bounded-parameter MDP (BMDP) abstraction to model the evolution of the stochastic system through discrete regions of the state space. Departing from the previous works, an optimal policy is computed over the BMDP abstraction to satisfy a high-level specification given in temporal logic. The framework constructs the entire abstraction and control policy from scratch, requiring only a model of the dynamics, a map of environment, and a task specification. Although errors are introduced when discretely approximating a continuous process, analysis shows that as the discrete regions become smaller, errors in the approximation limit to zero and the control policy that is computed converges to the true optimal.

This work presumes that the task specification is given in co-safe LTL [1], a subset of LTL. Although co-safe LTL has infinite semantics, a finite trace is sufficient to satisfy these formulas. In many robotics applications, tasks are required to be completed in finite time, making co-safe LTL an ideal language for such high-level specifications. A noteworthy property of the BMDP abstraction is that it can be reused for any co-safe LTL specification given the same robot and workspace. Simulated results show that given a BMDP abstraction, a complete control policy to satisfy the specification with maximum probability can be computed in seconds, orders of magnitude faster than existing techniques.

2 Problem Formulation

The objective of this work is to compute a control policy for a fully-observable robotic system with noisy actuation that satisfies a high-level task specification given in a fragment of LTL with maximal probability. Formal definitions of the robotic system, LTL specification language, and task satisfaction follow.

2.1 Stochastic Robotic System

Consider a robotic system with noisy actuation whose dynamics are described by the following stochastic differential equation [25, 12, 13, 24]:

$$dx = f(x(t), u(t))dt + F(x(t), u(t))dw, \qquad (1)$$
$$x \in X \subset \mathbb{R}^{n_x} \quad , \quad u \in U \subset \mathbb{R}^{n_u},$$

where X and U are compact sets representing the state and control spaces, and $w(\cdot)$ is an n_w-dimensional Wiener process. Functions $f : X \times U \to \mathbb{R}^{n_x}$ and $F : X \times U \to \mathbb{R}^{n_x \times n_w}$ are bounded and Lipschitz continuous, where $f(\cdot,\cdot)$ describes the robot's nominal dynamics and $F(\cdot,\cdot)$ captures the influence of noise on the dynamics. The pair $(u(\cdot), w(\cdot))$ is assumed to satisfy the Markov property. The stochastic process is fully observable and stops once the interior of X is left.

2.2 Syntactically Co-safe LTL

The mission of the stochastic system is specified by a syntactically co-safe LTL formula ϕ [1, 7]. The syntax and semantics of such a specification is given here for completeness.

Syntax: A syntactically co-safe LTL formula ϕ is defined inductively over a set $\Pi = \{\pi_1, \ldots, \pi_n\}$ of atomic Boolean propositions and a set of unary and binary operators:

$$\phi := \pi \,|\, \neg\pi \,|\, \phi \vee \phi \,|\, \phi \wedge \phi \,|\, \mathcal{X}\phi \,|\, \mathcal{F}\phi \,|\, \phi\,\mathcal{U}\phi,$$

where $\pi \in \Pi$ in an atomic proposition, $\neg$, $\vee$, and $\wedge$ represent the Boolean operators negation, disjunction, and conjunction respectively, $\mathcal{X}$ is the temporal *next* operator, $\mathcal{F}$ represents the temporal *eventually* operator, and $\mathcal{U}$ denotes the temporal *until* operator.

Semantics: The semantics of a syntactically co-safe LTL formula ϕ are defined over infinite traces of 2^{Π}. Let $\sigma = \{\tau_i\}_{i=0}^{\infty}$ denote an infinite *trace*, where $\tau_i \in 2^{\Pi}$. Furthermore, let $\sigma^i = \tau_i, \tau_{i+1}, \ldots$ denote a suffix of the trace starting at step i. The notation $\sigma \models \phi$ denotes that the trace σ satisfies co-safe formula ϕ and has the following recursive definition:

- $\sigma \models \pi$ if $\pi \in \tau_0$
- $\sigma \models \neg\pi$ if $\pi \notin \tau_0$
- $\sigma \models \phi_1 \vee \phi_2$ if $\sigma \models \phi_1$ or $\sigma \models \phi_2$
- $\sigma \models \phi_1 \wedge \phi_2$ if $\sigma \models \phi_1$ and $\sigma \models \phi_2$
- $\sigma \models \mathcal{X}\phi$ if $\sigma^1 \models \phi$
- $\sigma \models \mathcal{F}\phi$ if $\exists k \geqslant 0$ where $\sigma^k \models \phi$
- $\sigma \models \phi_1\mathcal{U}\phi_2$ if $\exists k \geqslant 0$ where $\sigma^k \models \phi_2$, and $\forall i \in [0, k), \sigma^i \models \phi_1$

Although the semantics have an infinite horizon, a *finite trace* is sufficient to satisfy ϕ. Thus, a deterministic finite automaton (DFA) $\mathcal{A}_\phi = (Z, \Sigma, \delta, z_0, T)$ can be constructed that accepts exactly the satisfying traces of ϕ, where

- Z is a finite set of states,
- $\Sigma = 2^{\Pi}$ is the input alphabet, where each input symbol is a truth assignment for all propositions in Π,
- $\delta : Z \times \Sigma \to Z$ is the transition function,
- $z_0 \in Z$ is the initial state, and
- $T \subseteq Z$ is the set of accepting states.

Let $\sigma = \sigma_0 \ldots \sigma_l$ be a string over Σ. $\mathcal{A}_\phi$ *accepts* σ iff a sequence of states $\omega_0 \ldots \omega_l$ exists in Z where $\omega_0 = z_0$, $\omega_{i+1} = \delta(\omega_i, \sigma_i)$ for $i = 0, \ldots, l-1$, and $\omega_l \in T$.

2.3 Stochastic Motion Planning with Temporal Goals

Stochastic system (1) evolves in a static workspace $\mathcal{W}$ consisting of a set of polytopic obstacles $\mathcal{O}$ and a set of polytopic regions $\mathcal{P} = \{p_1, \dots, p_n\}$, where p_i is mapped to atomic proposition π_i. Proposition π_i becomes true when the system visits any part of region p_i. With a slight abuse of notation, let σ denote the trajectory traced by the system during execution. Execution terminates when $\sigma \models \phi$ or $\sigma \cap \mathcal{O} \neq \emptyset$, whichever occurs first. Given these definitions, the problem addressed in this work is now formally stated:

Problem definition: Given a fully-observable stochastic system (1) operating in a workspace $\mathcal{W}$, compute a control policy for the system that maximizes the probability of satisfying a syntactically co-safe LTL formula ϕ.

3 Methodology

A top-down framework for computing an optimal control policy is presented in this section that maximizes the probability of satisfying a task specified in co-safe LTL. Computation of the policy occurs in two phases. First, the evolution of the stochastic system is abstracted to a particular kind of uncertain Markov model, a bounded-parameter Markov decision process (BMDP) $\mathcal{B}$. The BMDP models the range of transition probabilities that are observed when the system transitions between regions in a discretization of the state space. Second, the co-safe LTL specification ϕ is translated into an equivalent DFA $\mathcal{A}_\phi$, and the Cartesian product $\mathcal{P} = \mathcal{B} \times \mathcal{A}_\phi$ is computed. Conceptually, the product $\mathcal{P}$ is also a BMDP where each state is a unique tuple (q, z), where q is a discrete region of the state space and z is a state in $\mathcal{A}_\phi$. Then, an optimal policy is computed over $\mathcal{P}$ to reach any state (q', z'), where z' is accepting in $\mathcal{A}_\phi$. With respect to the discretization, an optimal policy over $\mathcal{P}$ satisfies ϕ with maximum probability. A block diagram illustrating the components of the planning framework is shown in Figure 1.

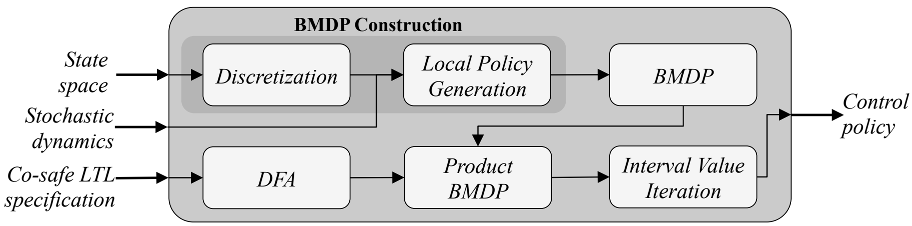


Fig. 1: Diagram of the proposed stochastic temporal logic planning framework.

3.1 BMDP Abstraction

To achieve computational tractability, the proposed framework abstracts the evolution of the stochastic system to motions between discrete regions of the state space. Since the system is stochastic, navigation of the system between any pair of adjacent regions is presumed to be imperfect. Furthermore, the probability of transitioning to an adjacent region depends on the initial state within the current region, which is not known *a priori*. Therefore, a range of transition probabilities is required to fully represent the likelihood of the system successfully moving between two regions, corresponding to the minimum and maximum over all initial conditions. The discretization, coupled with the transition probability ranges naturally lends itself to an *uncertain Markov decision process*. This particular construction of the region level abstraction, however, forms a special kind of uncertain MDP, known as a bounded-parameter MDP (BMDP) [23]. A BMDP is able to capture the uncertainty over the transition probabilities with a range of values, and can be solved optimally in polynomial time. In the remainder of this section, a formal definition of the BMDP is given, and the construction of the BMDP abstraction for stochastic planning is detailed.

Bounded-parameter MDP A bounded-parameter Markov decision process (BMDP) [23] is an MDP whose transition probabilities are not known exactly. Instead, these values are presumed to lie within a range of real numbers. Formally, a BMDP is a tuple $\mathcal{B} = (Q, A, \check{P}, \hat{P}, L)$, where

- Q is a finite set of states,
- A is a finite set of actions,
- $\check{P} : Q \times A \times Q \to [0, 1]$ and $\hat{P} : Q \times A \times Q \to [0, 1]$ are pseudo-transition probability functions that for state $q \in Q$ under action $a \in A$ return the minimum and maximum transition probabilities to state $q' \in Q$, respectively,
- $L : Q \to 2^{\Pi}$ is a labeling function that maps each $q \in Q$ to a set of atomic propositions in 2^{Π}. L relates discrete states with the proposition regions.

The following property must also hold in a BMDP: for all $q, q' \in Q$ and any $a \in A(q)$, $\check{P}(q, a, \cdot)$ and $\hat{P}(q, a, \cdot)$ are pseudo-distribution functions such that $0 \leqslant \check{P}(q, a, q') \leqslant \hat{P}(q, a, q') \leqslant 1$ and $\sum_{q' \in Q} \check{P}(q, a, q') \leqslant 1 \leqslant \sum_{q' \in Q} \hat{P}(q, a, q')$.

Discretization A discretization of the state space that respects both obstacles and proposition regions forms the states of the BMDP abstraction. Formally, a discretization of the bounded state space X is defined as a set of polytopic, non-overlapping subspaces of X whose union is X.

A desirable discretization depends on a number of factors, including the geometry and dynamics of the system. Practically speaking, the coarseness of the discretization has a direct impact on policy computation time. The difficulty of discretizing a high-dimensional space for motion planning purposes is well known [7]. The proposed framework advocates a discretization of the workspace using a Delaunay triangulation [26] that can easily be generated to respect obstacles and other regions of interest. Moreover, this triangulation avoids *skinny* triangles which may be deleterious to the abstraction. Note that discretizing the workspace induces a discretization of the state space by projecting each element of the state space into the workspace and identifying the region the projection lies in.

Local Policy Computation Given a discretization of the state space, a local controller or control policy is generated to optimally navigate the stochastic system between adjacent regions. These local policies correspond to the actions of the BMDP abstraction. The proposed framework is not dependent on a particular method for local policy generation, so long as the transition probability range for successfully moving between two regions can be calculated. A general method for computing local policies is the iMDP algorithm [12], a sampling-based approach that asymptotically approximates the optimal control policy for stochastic system (1) using a series of progressively larger Markov decision processes. When local policies are approximated with a Markov chain (as in iMDP), the minimum and maximum transition probabilities for transitioning to an adjacent discrete region are easily obtained with an *absorbing Markov chain* analysis [27]. The iMDP method is used to compute the local policies in the evaluation of this framework. Depending on the system employed, however, more specialized controllers can also be synthesized for stronger guarantees in the local control policies.

3.2 Product BMDP and Optimal Policy

Recall that the objective of the system is given as a co-safe LTL formula ϕ, and that a finite trace is able to satisfy this kind of specification. To compute a control policy to satisfy ϕ, the specification is first translated into an equivalent DFA [1]. Unfortunately, constructing $\mathcal{A}_\phi$ introduces an exponential blow-up with respect to the size of ϕ. Nevertheless, tools exist that emit a minimized DFA virtually instantly for the kinds of specifications commonly used for planning tasks [28]. Given $\mathcal{A}_\phi$, the product of $\mathcal{A}_\phi$ with the BMDP described above is computed, and then a policy over the product is obtained to satisfy the specification with maximum probability. The product BMDP is formally defined below.

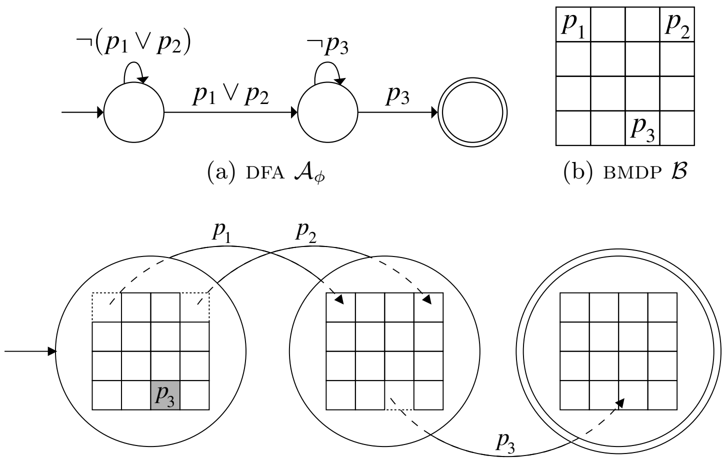
Product BMDP Given a BMDP $\mathcal{B}$ and a DFA $\mathcal{A}_\phi$ for a co-safe LTL specification ϕ, the product BMDP $\mathcal{P} = \mathcal{B} \times \mathcal{A}_\phi$ is a tuple $\mathcal{P} = (Q_\mathcal{P}, T_\mathcal{P}, A_\mathcal{P}, \check{P}_\mathcal{P}, \widehat{P}_\mathcal{P})$, where

$$Q_\mathcal{P} = Q \times Z, \qquad T_\mathcal{P} = Q \times T, \qquad A_\mathcal{P} = A,$$

$$\check{P}_\mathcal{P}((q,z), a_\mathcal{P}, (q',z')) = \begin{cases} \check{P}(q,a,q') & \text{if } z' = \delta(z, L(q')) \\ 0 & \text{otherwise,} \end{cases}$$

$$\widehat{P}_\mathcal{P}((q,z), a_\mathcal{P}, (q',z')) = \begin{cases} \widehat{P}(q,a,q') & \text{if } z' = \delta(z, L(q')) \\ 0 & \text{otherwise,} \end{cases}$$

for $q, q' \in Q$, $a_\mathcal{P} \in A_\mathcal{P}$, $a \in A$, and $z, z' \in Z$. Conceptually, $\mathcal{P}$ is both a BMDP and a DFA. The goal is to compute a policy over the actions $\mathcal{A}_\mathcal{P}$ in $\mathcal{P}$ to reach any terminal state $(q,z) \in T_\mathcal{P}$ with maximum probability. Note that transitions in the BMDP component of each state still obey the transition probabilities over the actions between each discrete region, and a transition in the DFA occurs only when the system enters a labeled proposition region that has a transition in the current DFA state. Therefore, the policy that maximizes the probability of reaching a state in $T_\mathcal{P}$ optimizes the probability of satisfying ϕ (reaching an accepting state in $\mathcal{A}_\phi$). A conceptual illustration of the of the product BMDP $\mathcal{P}$ given $\mathcal{B}$ and $\mathcal{A}_\phi$ is shown in Figure 2.



(a) DFA $\mathcal{A}_\phi$

(b) BMDP $\mathcal{B}$

(c) A conceptual illustration of the product BMDP $\mathcal{P} = \mathcal{B} \times \mathcal{A}_\phi$

Fig. 2: (a) The minimal DFA $\mathcal{A}_\phi$ for $\phi = (\neg p_3 \,\mathcal{U} (p_1 \vee p_2)) \wedge \mathcal{F} p_3$. (b) A discretization of the state space, allowing for the construction of a BMDP $\mathcal{B}$ with proposition regions p_1, p_2, and p_3. (c) An illustration of the product BMDP $\mathcal{P} = \mathcal{B} \times \mathcal{A}_\phi$. Specification ϕ requires the system to transition through $\mathcal{P}$ by visiting regions p_1 or p_2, followed by region p_3. If proposition p_3 is visited first, ϕ cannot be satisfied. The accepting state is denoted with the double circle.

Optimal Policy Computation Finding a policy over $\mathcal{P}$ to satisfy specification ϕ is equivalent to solving the *maximal reachability probability problem* [29]. The objective in this problem is to find the maximum probability that a set of states can be reached from any other state in an MDP. Prior work also solves the maximal reachability probability problem for a BMDP [30]. The key difference for a BMDP is that the expected value (maximum probability) for each state is not a scalar value, but rather a range derived from the transition probability bounds.

Note that a BMDP represents a uncountably-large set of MDPs whose transition probabilities lie in those of the BMDP. This implies that the optimization objective for a BMDP is ambiguous since the true probabilities are unknown. The literature proposes two optimal policies: a *pessimistic* policy that optimizes for the lower bound probabilities, and an *optimistic* policy that optimizes for the upper bound probabilities [23]. From these two criteria, absolute optimal value ranges for each state in the BMDP naturally correspond to the minimum pessimistic value and the maximum optimistic value.

The algorithm for computing an optimal policy in a BMDP is *interval value iteration* (IVI), the analog of *value iteration* for an MDP. Before IVI begins, an optimization objective for the BMDP must be chosen (e.g., pessimistic or optimistic). For each iteration of IVI, an MDP representative is selected, based on the optimization objective and the current value estimate, and the typical Bellman backup is computed. Let $\widetilde{P}$ denote the probability distribution for the MDP representative selected during an iteration of IVI for the product BMDP

$\mathcal{P}$. Then the Bellman backup operation for computing the maximum reachable probabilities in $\mathcal{P}$ is:

$$v(q) = \begin{cases} 1 & \textit{if } q \in T_{\mathcal{P}} \\ \max_{a \in A(q)} \left[\sum_{q' \in Q} \widetilde{P}(q, a, q') v(q') \right] & \text{otherwise.} \end{cases} \quad (2)$$

The result of the interval value iteration computation (2) is a control policy that maximizes the probability of satisfying the co-safe LTL specification ϕ over the BMDP abstraction. The value $v(q)$ represents the probability that the stochastic system, starting anywhere in region q, reaches an accepting state in the automaton $\mathcal{A}_{\phi}$. Since IVI reasons over discrete regions of the state space rather than individual elements, significant savings in computation time are realized.

4 Analysis

This section analyzes the asymptotic convergence of the probability of satisfying a co-safe LTL specification ϕ computed over the BMDP abstraction to the true optimal values for stochastic system (1). It is shown that the BMDP approximates the continuous dynamics with a bounded error that is a function of the diameter of each polytopic region. As the largest diameter in the discretization shrinks to zero, uncertainty in the optimal value estimates for the BMDP are eliminated, indicating convergence to the true maximum probabilities for the system to satisfy ϕ. Proof of these claims begins by inspecting the local policies of the BMDP. A typical method for computing such policies uses a discrete, *locally consistent* approximation of the continuous dynamics, defined below.

Definition 4.1 (Definition 1.3 in [25]). *Let ξ denote a controlled Markov chain approximating a stochastic system* (1) *whose dynamics are given by bounded, Lipschitz continuous functions f and F. Each state $x \in \xi$ is associated with a non-negative holding time $\Delta t(x)$, representing the time a control u is applied at state x. Let ξ_i denote the i^{th} state resulting from the stochastic process ξ, and the notation $\Delta\xi_i = \xi_{i+1} - \xi_i$ denote the distance between two consecutive states in the discrete approximation. A discrete time Markov chain ξ is locally consistent with continuous-time system* (1) *if the following conditions are met for all $x \in \xi$, where $w \in U$ is the control applied at state x:*

$$\mathbb{E}[\Delta\xi_i | \xi_i = x, u_i = w] = f(x, w)\Delta t(x) + O(\Delta t(x)) \quad (3)$$

$$Cov[\Delta\xi_i | \xi_i = x, u_i = w] = F(x, w)F(x, w)^T \Delta t(x) + O(\Delta t(x)) \quad (4)$$

where $O(\cdot)$ indicates an upper bound on the error introduced by the discrete time approximation of the continuous dynamics as a function of the holding time.

In the BMDP abstraction, actions for each discrete region (local policies) are presumed to be locally consistent Markov chains of stochastic system (1). Note, the iMDP method [12] computes a locally consistent Markov chain. A transition between regions in the BMDP, however, likely requires a series of discrete time steps to complete. Since each action is locally consistent, the modeling error in each BMDP transition is bounded, as shown in the following lemma.

Lemma 4.2. *Given a* BMDP *abstraction of stochastic system* (1) *where actions induce locally consistent Markov chains, the error incurred by a transition from region q to adjacent region q' is bounded by the maximum expected time to exit q.*

Proof. Let ξ^μ denote the locally consistent Markov chain induced by action μ in the BMDP abstraction defined over q that attempts to navigate the system from q to an adjacent region q'. Furthermore, let $\Delta T_x(\xi^\mu)$ be the expected time for the system to exit region q from initial state $x \in \xi^\mu$. From (3),(4), the error introduced by ξ^μ is bounded by the discrete holding times at each state in ξ^μ. It then follows directly that the error in the transition from region q is bounded by $\max_{x \in \xi^\mu} O(\Delta T_x(\xi^\mu))$, which is the maximum error that accumulates when the system evolves within q under μ over all possible initial states. □

Furthermore, the expected exit time for system (1) from a bounded region is always finite, and this time is a function of the initial state and the diameter of the region ([31], Chapter III, Lemma 3.1). Given the error incurred by the BMDP abstraction of system (1) as a function of the diameter of each region, what remains to prove is that as the maximum diameter shrinks to zero, an optimal BMDP policy asymptotically converges to an optimal policy for the continuous system. Arguments are based on the value functions corresponding to the optimal policies, and begin by inspecting the transition probability ranges in the BMDP. For convenience, $diam(q)$ denotes the diameter of a polytopic region q in the discretization.

Lemma 4.3. *Let μ denote a locally optimal, locally consistent control policy that navigates the system* (1) *from region q to a region adjacent to q in a* BMDP *abstraction. Then, for all q' adjacent to q:*

$$\lim_{diam(q) \to 0} \left[\hat{P}(q, \mu, q') - \check{P}(q, \mu, q') \right] = 0. \tag{5}$$

Proof (sketch). The Lipschitz assumption for stochastic system (1) asserts $||f(x,u) - f(x',u')|| \leqslant K(||x - x'|| + ||u - u'||)$, where $K \in \mathbb{R}$ is the Lipschitz constant. An analogous assertion also holds for the covariance F. Since the system evolves according to a locally optimal policy μ to maximize the probability of reaching an adjacent, contiguous region, it follows from the Lipschitz condition of f, F that the optimal transition probabilities for two states x, x' in a discrete region q differ only by a function of the distance between x and x'. As the diameter of q shrinks to zero, the maximum distance between any two states in q also decreases to zero, indicating that the transition probability ranges under μ to reach all neighboring regions also converge to scalar values. □

For any policy over a BMDP, the range of optimal value function estimates falls within the minimum pessimistic value and the maximum optimistic value. The following lemma shows that these policies and value function estimates always exist. The subsequent theorem then relates the value function estimate to the continuous dynamics (1), showing that the values converge to the maximum probability of satisfying a co-safe LTL specification for each state in the product BMDP abstraction as largest diagonal in the discretization approaches zero.

Lemma 4.4. *(Theorems 8 and 9 in [23]) For any* BMDP *there exists an optimistically optimal and a pessimistically optimal policy. These policies converge pointwise to the desired optimal value function.*

Theorem 4.5. *Let $\check{v}(q_p)$ denote the minimum pessimistically optimal value and $\hat{v}(q_p)$ denote the maximum optimistically optimal value for a state q_p computed by* (2) *over the product* BMDP *abstraction $\mathcal{P}$ for the stochastic system* (1) *and co-safe specification ϕ. Then, for all $q_p \in Q_{\mathcal{P}}$:*

$$\lim_{\max_{q \in Q} diam(q) \to 0} [\hat{v}(q_p) - \check{v}(q_p)] = 0, \tag{6}$$

and $\hat{v}(q_p) = \check{v}(q_p)$ is the maximum probability of satisfying ϕ for all states $x \in q_p$ of stochastic system (1)*.*

Proof (sketch). It follows directly from Lemmas 4.3 and 4.4 that the value function range for each region in the BMDP abstraction must converge to a single value as the diameter of the largest discrete region shrinks to zero. Thus, (6) holds. Furthermore, from Lemma 4.2, the BMDP models the underlying dynamics of the continuous stochastic system arbitrarily well as the largest diameter in the discretization shrinks to zero. Therefore, as $|\hat{v}(q_p) - \check{v}(q_p)|$ approaches 0 for all states in the product BMDP, the value range for q_p converges to a scalar value that is the continuously optimal value for all states $x \in q_p$. □

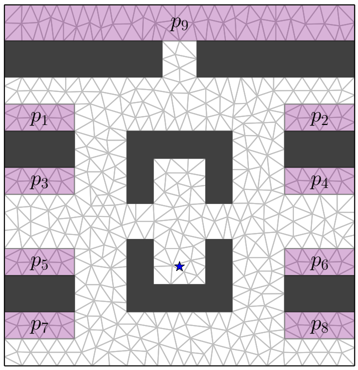
5 Evaluation

Evaluation of the proposed method for computing a control policy that satisfies specification ϕ with maximum probability is given in this section. A 2D system with single integrator dynamics and Gaussian noise is simulated. Formally, $f(x, u) = u$ and $F(x, u) = 0.1I$, where I is the identity matrix, as in [12, 13, 24]. Computations are performed on a 2.4GHz Intel Xeon CPU with 12GB memory.

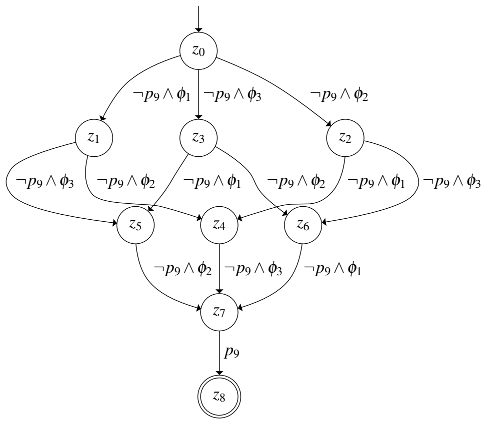
Simulated experiments are performed in a 20×20 *warehouse* inspired environment, shown in Fig. 3a. A set of proposition regions, $p_1, \ldots, p_8$, represent regions of interest in the warehouse, and region p_9 represents a processing station where completed orders are taken. Two different co-safe LTL specifications are evaluated. The first specification, ϕ_G, represents a *gathering* task, where the system must retrieve three items in any order, then bring the completed order to the processing station. Since the same item could exist in multiple locations, subformulas $\phi_1 = (p_1 \vee p_3)$, $\phi_2 = (p_2 \vee p_4)$ and $\phi_3 = (p_5 \vee p_6 \vee p_7 \vee p_8)$ denote the possible locations for items 1, 2, and 3, respectively. The second task, ϕ_S, is a rigid *sequence* of items to gather, where item 1 must be retrieved before item 2, and item 2 must be retrieved before item 3, and only then may the system return to the processing station. ϕ_G and ϕ_S are represented in co-safe LTL as:

$$\phi_G = (\neg p_9 \,\mathcal{U} \phi_1) \wedge (\neg p_9 \,\mathcal{U} \phi_2) \wedge (\neg p_9 \,\mathcal{U} \phi_3) \wedge \mathcal{F} p_9$$
$$\phi_S = \mathcal{F}(\phi_1 \wedge \mathcal{X}\mathcal{F}(\phi_2 \wedge \mathcal{X}\mathcal{F}(\phi_3 \wedge \mathcal{X}\mathcal{F} p_9))).$$

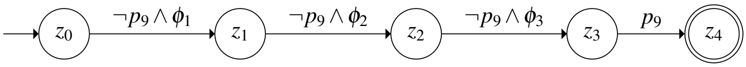
The minimized automata for ϕ_G and ϕ_S are shown in Figures 3b and 3c.



(a) Environment and discretization



(b) Minimized automaton $\mathcal{A}_{\phi_G}$



(c) Minimized automaton $\mathcal{A}_{\phi_S}$

Fig. 3: (a) The 20×20 *warehouse*. Obstacles are gray, and nine proposition regions are shaded and labeled. An obstacle and proposition respecting triangulation (826 triangles) is overlayed. The system starts at the star. (b) Minimized DFA for ϕ_G. (c) Minimized DFA for ϕ_S. Self-transitions in the DFAs are omitted for clarity.

The computation time and quality of the resulting control policy from the proposed BMDP abstraction are evaluated here. To compare this work against existing methods for planning under uncertainty, two state-of-the-art frameworks are extended to compute policies that satisfy a co-safe LTL specification. The first method employs a typical MDP abstraction, constructed using the SMR method [11]. Eight unit controls spanning the cardinal and ordinal directions are applied in the SMR for a fixed duration of 100ms. Given the SMR, a policy is computed over the product of the SMR with $\mathcal{A}_\phi$ in the style of existing temporal logic methods. The second approach utilizes the iMDP algorithm [12] and iteratively constructs an optimal policy directly in the continuous state space $X \times \mathcal{A}_\phi$. In the BMDP abstraction, a discretization with 826 triangles is used, local policies are computed using iMDP, and a pessimistically optimal policy over the BMDP is computed. All three methods are executed until there are 750 states sampled per unit area or four hours elapses, whichever is first. Previous work has shown this sampling density yields favorable policies for the system evaluated [13].

Discrete abstraction The first step for the BMDP and SMR methods is to construct a discrete abstraction that models the evolution of the stochastic system in the environment. This construction can be thought of as a one-time cost since the abstraction can be reused for different tasks, provided the environment and robot stay the same. The iMDP algorithm does not emit a reusable abstraction since an optimal policy is constructed directly by this method. Table 1 shows that constructing the BMDP abstraction (over the discretization in Fig. 3a) is

	Abstraction time (s)	Policy time (s)		Prob. Success	
		ϕ_G	ϕ_S	ϕ_G	ϕ_S
BMDP	**1181.59**	**10.80**	**5.87**	0.979	0.973
SMR	2494.68	1728.56	1234.97	**1.000**	**1.000**
iMDP	n/a	14400.00	14400.00	0.899	0.971

Table 1: The average time to generate the discrete abstractions, average policy computation time, and median probability of success for tasks ϕ_G and ϕ_S in the three methods evaluated. All values are taken over 50 independent runs. The abstraction for each method is a one time cost, and can be reused for any ϕ.

significantly faster than a comparable MDP abstraction. BMDP construction for 826 discrete regions takes less than 20 minutes on a single processor, compared to over 45 minutes for SMR.

Policy computation Computing a policy to satisfy the specification with maximum probability exposes stark differences in the three different methods, as noted in Table 1. In the BMDP and SMR methods, the Cartesian product of the Markov abstraction is taken with the automaton $\mathcal{A}_\phi$, and an optimal policy over this product is computed. For iMDP, the policy is computed directly in the product space. The BMDP abstraction requires just over 10 seconds to find an optimal (pessimistic) policy for ϕ_G and under 6 seconds to find an optimal policy for ϕ_S. Compare these times to SMR, which requires nearly 30 minutes for ϕ_G and over 20 minutes for ϕ_S. This difference accentuates the gains in reasoning over discrete regions rather than individual state space elements. The iMDP method consistently reached a four hour timeout, and only contains about half of the number of discrete states that exist the final BMDP and SMR policies; the complexity of iMDP depends on the number of states in the existing approximating structure, where each iteration takes more time than the previous.

Probability of Success Naturally, the significant gains in computation time for the BMDP abstraction do not come without a price. The last two columns in Table 1 show the median probability of success to satisfy each of the specifications across all three methods. Although the SMR abstraction does not provide any formal guarantees, this method is able to consistently find a virtually perfect policy. This result can be attributed to the relatively simple system evaluated coupled with the rather dense MDP abstraction utilized. Nevertheless, the much coarser BMDP abstraction cedes only 2-3% probability of success compared to SMR while providing computation times that are substantially faster. Although iMDP provides strong theoretical guarantees, the complexity of this method prohibits scalability into the large product state space. This is particularly evident for ϕ_G, where $\mathcal{A}_\phi$ has 9 states, and iMDP has a probability of success at just around 90%.

6 Discussion

This work presents a method for efficient stochastic motion planning where the objective is a high-level specification given in co-safe LTL. By abstracting the evolution of the robot to a bounded-parameter MDP where the states are discrete regions of the state space, the method is able to quickly and effectively compute an optimal policy over the product of the BMDP abstraction and a DFA representing the high-level specification with maximum probability. Evaluation of the approach shows that policies for co-safe LTL specifications can be obtained in seconds once an abstraction is constructed. The BMDP abstraction admits optimal policy computation that is orders of magnitude faster than existing methods.

The analysis of the method indicates that as the discretization becomes finer, errors introduced in the BMDP abstraction model limit to zero and the policy asymptotically converges to optimal. As presented, the framework does not actively seek to reduce the transition probability ranges or discrete region sizes to achieve asymptotic optimality directly. It is a natural extension of this work, however, to refine local policies with large probability ranges by shrinking the discrete region they are defined over.

The relatively simple dynamics considered in the evaluation of this work should not be considered a limiting factor. The dynamics are reasoned over only at the BMDP abstraction level. For a more complex system, the time to compute the BMDP abstraction will surely increase, but time to computing the satisfying policy is polynomial in the number of discrete regions.

Acknowledgements Work by Ryan Luna is supported by a NASA Space Technology Research Fellowship. Work by Morteza Lahijanian, Mark Moll, and Lydia Kavraki is supported in part by NSF NRI 1317849, NSF 1139011, and NSF CCF 1018798. Computing resources supported in part by NSF CNS 0821727.

References

1. Kupferman, O., Vardi, M.Y.: Model checking of safety properties. Formal Methods in System Design **19**(3) (2001) 291–314
2. Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics. MIT Press (2005)
3. Fainekos, G.E., Kress-Gazit, H., Pappas, G.J.: Temporal logic motion planning for mobile robots. In: IEEE Int'l. Conf. on Robotics and Automation. (2005) 2020–2025
4. Gazit, H.K., Fainekos, G., Pappas, G.J.: Where's waldo? Sensor-based temporal logic motion planning. In: IEEE Int'l. Conf. on Robotics and Automation. (2007) 3116–3121
5. Kloetzer, M., Belta, C.: A fully automated framework for control of linear systems from temporal logic specifications. IEEE Trans. on Automatic Control **53**(1) (2008) 287–297
6. Wongpiromsarn, T., Topcu, U., Murray, R.M.: Receding horizon control for temporal logic specifications. In: Int'l Conf. on Hybrid Systems: Computation and Control. (2010) 101–110
7. Bhatia, A., Kavraki, L., Vardi, M.: Motion planning with hybrid dynamics and temporal goals. In: IEEE Conf. on Decision and Control. (2010) 1108–1115
8. Bhatia, A., Maly, M., Kavraki, L., Vardi, M.: Motion planning with complex goals. IEEE Rob. and Autom. Magazine **18**(3) (2011) 55–64
9. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning with deterministic μ-calculus specifications. In: Am. Control Conf. (2012) 735–742
10. Plaku, E., Kavraki, L.E., Vardi, M.Y.: Falsification of LTL safety properties in hybrid systems. Software Tools for Technology Transfer **15**(4) (2013) 305–320

11. Alterovitz, R., Siméon, T., Goldberg, K.: The stochastic motion roadmap: A sampling framework for planning with Markov motion uncertainty. In: Robotics: Science and Systems. (2007) 246–253
12. Huynh, V.A., Karaman, S., Frazzoli, E.: An incremental sampling-based algorithm for stochastic optimal control. In: IEEE Int'l. Conf. on Robotics and Automation. (2012) 2865–2872
13. Luna, R., Lahijanian, M., Moll, M., Kavraki, L.E.: Fast stochastic motion planning with optimality guarantees using local policy reconfiguration. In: IEEE Int'l. Conf. on Robotics and Automation. (2014) 3013–3019
14. Clarke, E.M., Grumberg, O., Peled, D.: Model checking. MIT Press (1999)
15. Kress-Gazit, H., Wongpiromsarn, T., Topcu, U.: Correct, reactive robot control from abstraction and temporal logic specifications. IEEE Rob. and Autom. Magazine **18**(3) (2011) 65–74
16. Ding, X.C., Kloetzer, M., Chen, Y., Belta, C.: Formal methods for automatic deployment of robotic teams. IEEE Rob. and Autom. Magazine **18**(3) (2011) 75–86
17. DeCastro, J.A., Kress-Gazit, H.: Guaranteeing reactive high-level behaviors for robots with complex dynamics. In: IEEE/RSJ Int'l. Conf. on Intelligent Robotics and Systems. (2013) 749–756
18. Vasile, C., Belta, C.: Sampling-based temporal logic path planning. In: IEEE/RSJ Int'l. Conf. on Intelligent Robotics and Systems. (2013) 4817–4822
19. Maly, M.R., Lahijanian, M., Kavraki, L.E., Kress-Gazit, H., Vardi, M.Y.: Iterative temporal motion planning for hybrid systems in partially unknown environments. In: Int'l Conf. on Hybrid Systems: Computation and Control. (2013) 353–362
20. Ding, X.C., Smith, S.L., Belta, C., Rus, D.: MDP optimal control under temporal logic constraints. In: IEEE Conf. on Decision and Control. (2011) 532–538
21. Lahijanian, M., Andersson, S.B., Belta, C.: Temporal logic motion planning and control with probabilistic satisfaction guarantees. IEEE Trans. on Robotics **28**(2) (2012) 396–409
22. Wolff, E.M., Topcu, U., Murray, R.M.: Robust control of uncertain Markov decision processes with temporal logic specifications. In: IEEE Conf. on Decision and Control. (2012) 3372–3379
23. Givan, R., Leach, S., Dean, T.: Bounded-parameter Markov decision processes. Artificial Intelligence **122** (2000) 71–109
24. Luna, R., Lahijanian, M., Moll, M., Kavraki, L.E.: Optimal and efficient stochastic motion planning in partially-known environments. In: AAAI Conf. on Artificial Intelligence. (2014)
25. Kushner, H.J., Dupuis, P.: Numerical methods for stochastic control problems in continuous time. Volume 24. Springer (2001)
26. Shewchuk, J.R.: Delaunay refinement algorithms for triangular mesh generation. Comp. Geometry **22**(1-3) (2002) 21–74
27. Kemeny, J.G., Snell, J.L.: Finite Markov Chains. Springer-Verlag (1976)
28. Duret-Lutz, A., Poitrenaud, D.: Spot: an extensible model checking library using transition-based generalized Büchi automata. In: IEEE/ACM Int'l Symp. on Modeling, Analysis, and Simulation of Comp. and Telecom. Systems. (2004) 76–83
29. de Alfaro, L.: Formal Verification of Probabilistic Systems. PhD thesis, Stanford University (1997)
30. Wu, D., Koutsoukos, X.: Reachability analysis of uncertain systems using bounded-parameter Markov decision processes. Artificial Intelligence **172**(8-9) (2008) 945–954
31. Freidlin, M.: Functional Integration and Partial Differential Equations. Princeton University Press (1985)