Evaluating Policy in the Context of Autonomous Vehicles in the Uncertain Environment

Md Rayhanul Islam

1 Introduction

Policy synthesis plays an important role in stochastic dynamic systems where system behavior is modeled by the Markov Decision Process (MDP) or Markov chain (MC). The primary objective of this synthesis process is to derive an optimal policy to satisfy the specification usually provided in linear temporal logic (LTL). However, modeling the system behavior using MDP/MC requires the transition probability to be known exactly. In many cases, knowing the exact system transition probability is unrealistic because external factors control system behavior.

In order to deal with the unknown system transition probability, an uncertain MDP (uMDP) is used. In a uMDP, the system transition probability and rewards associated with different states and actions lie in an uncertain set, which leads to ambiguity in the system behavior. The existing work focuses on either identifying whether all parameter valuations satisfy the specification [1] or computing how much the model satisfies the specification [2]. However, in the context of autonomous vehicles, knowing the parameter values for which the specification is satisfied is less relevant than knowing whether the current policy satisfies the specification for all possible valuations of the parameters. Furthermore, autonomous vehicles need a policy to satisfy the safety property no matter the situation.

In this project, we work on evaluating a policy in an uncertain environment where agents consider other agents' models as uMDP. Unlike other work, we assume autonomous vehicles start with an initial policy based on historical data. This assumption is realistic because autonomous vehicles must have a policy if it is running. Then our approach analyzes the initial policy to see if it is safe for all valuation of parameters in the parameter space of the uncertain environment. If the initial policy is unsafe, it presents an approach to finding a new policy that meets the safety property for all valuations of the parameter space. Finally, if the initial policy and the new policy could not satisfy the safety property, it suggests updating the model of the autonomous vehicles so that it satisfies the specification in the given scenario.

1.1 Contributions

The contribution of this work is provided below:

1. It computes whether the initial policy satisfies the safety property for all valuations of the parameter space in the uncertain model.
2. It provides an approach to compute a new policy if it exists so that it satisfies the safety property for all valuations of the parameter space in the uncertain model.
3. In case the initial and the new policy cannot satisfy the safety property, it provides an approach to update the system behavior of the agent so that the initial policy satisfies the safety property for all valuations of the parameter space in the uncertain model.

1.2 Preliminaries

The probability distribution over a finite discrete set X is a function $\mu : X \to [0,1] \subseteq \mathbb{R}$ such that $\sum_{x \in X} \mu(x) = 1$. Consider $V = \{x_1, \ldots, x_n\}$ is a finite set of variables (parameters) over $\mathbb{R}^n$. The set of polynomials over V with rational coefficients is denoted as $\mathbb{Q}[V]$. The cardinality of a set X is defined as $|X|$.

Markov Chain. A Markov chain (MC) [3] is a tuple (S, s_0, P, AP, L), where S is a set of states, $s_0 \in S$ is the initial state, $P : S \times S \to [0,1]$ is a transition function such that for any state $s \in S$, $\sum_{s' \in S} P(s, s') = 1$, AP is a set of atomic propositions and $L : S \to 2^{AP}$ is a labeling function. An MC is called finite if S and AP are finite.

Markov Decision Process. A Markov Decision Process (MDP) [3] is a tuple (S, s_0, Act, P, AP, L), where S, s_0, AP, and L are defined as in the definition of an MC, Act is a set of actions and $P : S \times Act \times S \to [0,1]$ is a transition probability function such that for any state $s \in S$ and action $\alpha \in Act$, $\sum_{s' \in S} P(s, s') \in \{0, 1\}$. An MDP is finite if S, Act, and AP are finite.

Parametric Markov Decision Process. A parametric Markov Decision Process (pMDP) [2] is a tuple $M = (S, Act, s_0, V, P, AP, L)$ where S, s_0, AP, and L are defined as in the definition of an MDP. V is the finite set of real-valued variables, and the transition function $P : S \times Act \times S \to \mathbb{Q}[V]$. Here, we restricted $\mathbb{Q}[V]$ to be a set of all polynomials over the variable set V. We intentionally considered $\mathbb{Q}[V]$ polynomials here; however, any algebraic expression could be used as long as it can form MDP. A pMDP is an MDP if the transition function yields a well-defined probability distribution such that for $s \in S$ and $\alpha_1 \in Act$, $P : S \times Act \times S \to [0, 1]$ and $\sum_{s' \in S} P(s, \alpha, s') = 1$.

The parameter space of M is defined as V_M. An instantiation $u \in V_M$ to a pMDP M forms an instantiated MDP $M[u]$ by replacing each $f \in \mathbb{Q}[u]$ in M by $f[u]$. An u is well defined for M if $M[u]$ becomes an MDP.

We also define $ActS : S \to A$, which provides the set of enabled actions in a state. For $s \in S$, $ActS(s)$ is defined as $ActS(s) = \{\alpha \in A \mid \exists s' \in S, P(s, \alpha, s') \neq 0\}$. In addition, for $s \in S$, the function $Post : S \to S$ is defined as $Post(s) = \{s' \in S \mid \exists \alpha \in A, P(s, \alpha, s') \neq 0\}$ which returns the set of all direct successors of state s. In this context, we assumed that all parameter instantiations in V_M yield well-defined MDPs. A pMDP becomes a parametric MC (pMC) if $ActS(s) = 1$ for all $s \in S$.

Uncertain Markov Decision Process. An uncertain Markov Decision Process (uMDP) [2] $M_{\mathbb{P}}$ is a tuple $M_{\mathbb{P}} = (M, \mathbb{P})$ where M is a pMDP and $\mathbb{P}$ is a probability distribution over the parameter space V_M. In other words, a uMDP is a pMDP with an associated distribution over possible parameter instantiations. Therefore, a realization of $\mathbb{P}$ yields a concrete MDP $M[u]$ with the respective instantiation $u \in V_M$. If M is a pMC, then $M_{\mathbb{P}}$ becomes an uncertain MC (uMC).

Path [3]. A finite path π of an (instantiated) MDP M is a sequence of states $s_0 s_1, \ldots, s_n$ such that $s_i \in Post(s_{i-1})$ for all $0 < i \leq n$ and $n \geq 0$. $Last(\pi)$ is the last state of π, and the set of all finite paths of M is $Paths^M_{fin}$.

Policy [3]. A policy σ for an (instantiated) MDP M is a function $\sigma : Paths^M_{fin} \to A$ where $\sigma(\pi) \in ActS(Last(\pi))$ for all $\pi \in Paths^M_{fin}$. The set of all policies is of Σ. In this considered problem, we assume the policies are memoryless.

Induced Markov Chain [3, 2, 4]. Consider an MDP M and a policy σ, then the MC induced by M and σ is given by $M^\sigma = (Paths^M_{fin}, A, s_0, P^\sigma, AP, L^\sigma)$ where

$$P^\sigma(\pi, \pi') = \begin{cases} P(Last(\pi), \sigma(\pi), s'), & \text{if } \pi' = \pi s' \\ 0, & \text{otherwise} \end{cases}$$

$L^\sigma(\pi) = L(Last(\pi))$, and the initial state s_0 and atomic proposition AP of M^σ remain same with M.

Product MDP. Let's consider two (instantiated) MDP $M_1 = (S_1, A_1, s_{1_0}, P_1, AP_1, L_1)$ and

$M_2 = (S_2, A_2, s_{2_0}, P_2, AP_2, L_2)$. The parallel composition of M_1 and M_2 is defined as $M_1 \parallel M_2 = (S_1 \times S_2, A_1 \times A_2, s_{1_0} \times s_{2_0}, P, AP_1 \cup AP_2, L)$, where $L(s_1 \times s_2) = L(s_1) \cup L(s_2)$ [4]. For any two state (s_1, s_2) and (s'_1, s'_2), and action $(\alpha_1, \alpha_2) \in A_1 \times A_2$, the transition function $P : S_1 \times S_2 \times A_1 \times A_2 \times S_1 \times S_2 \to [0, 1]$ is defined as $P((s_1, s_2), (\alpha_1, \alpha_2), (s'_1, s'_2)) = P_1(s_1, \alpha_1, s'_1) \times P_2(s_2, \alpha_2, s'_2)$ if $P_1(s_1, \alpha_1, s'_1) > 0$ and $P_2(s_2, \alpha_2, s'_2) > 0$.

2 Related work

In recent years, the uncertainties in MDPs have received significant momentum in the control and planning literature. This paper [5] analyzed the Markov models with uncertain rewards. It uses statistical methods to compute the likelihood of an MDP satisfying the cost requirement. However, this method only finds the reward parameters and does not calculate confidence intervals of satisfying a cost specification. Another work [1] explains the parameter synthesis problem for parametric MC/MDP by formulating the problems as the product MDP and separating the parameter space into regions based on satisfying the specification. However, when multiple agents interact, the system suffers from state explosion problems.

The above state explosion problem is solved by [6]. This approach proposes an incremental approach to synthesize control policies for a non-independent heterogeneous multi-agents system to maximize the probability of satisfying the specification; additionally, to tackle the state explosion problem, it initially incorporates a small subset of agents in the synthesis procedure and then continues to add more agents until the limitation of computational resources is reached.

This research [7] verifies a given specification in an MDP with uncertainties. It considers the uncertainty set for different states in an MDP to be independent. However, in many applications uncertain set is not independent. This research [2] proposes a sampling-based approach to approximately calculate the probability for any randomly drawn sample that satisfies an LTL specification. However, Unlike [1], which looks at whether all (or some) parameter values satisfy a specification, [2] concentrates on computing how much the model satisfies the specification.

3 Problem Formulation

Consider a set of agents $\mathbb{A} = \{A_1, A_2, \ldots, A_n\}$ are interacting with each other. The true behavior of $\{A_1, A_2, \ldots, A_n\}$ are modeled by MDP $M_T^1, M_T^2, \ldots, M_T^n$ respectively, and the true complete system $S_T = M_T^1 \parallel M_T^2 \parallel \ldots \parallel M_T^n$. All agents need to satisfy the specification φ. Although A_i has access to its true model M_T^i, due to the uncertainty in the environment, A_i does not know the true model of other agents $\mathbb{A} \setminus A_i$. The initial policy of A_i is σ_{init}, which comes from the historical data. Agent A_i may need to update its policy whenever A_i has a change in the observation regarding other agent's $\mathbb{A} \setminus A_i$'s model because policy from the historical data may not work. Due to uncertainty, A_i models other agents $\mathbb{A} \setminus A_i$ as the uMDP/uMC $\{\mathcal{M}^1_{\mathbb{P}^i_1}, \mathcal{M}^2_{\mathbb{P}^i_2}, \ldots, \mathcal{M}^n_{\mathbb{P}^i_n}\}$, where $\mathcal{M}^j_{\mathbb{P}^i_j} = (M^i_j, \mathbb{P}^i_j)$ for $j \in \{1, \ldots, n\}$, and pMDP/pMC $M^i_j = (S_j, Act, s_{j_0}, V^i_j, P^i_j, AP, L)/(S_j, s_{j_0}, V^i_j, P^i_j, AP, L)$, and $\mathbb{P}^i_j$ is a probability distribution over the parameter space $V_{M^i_j}$. We consider the true behavior of each agent remains in the uMDP/uMC such as $M_T^1 \in \mathcal{M}^1_{\mathbb{P}_i}$, $M_T^2 \in \mathcal{M}^2_{\mathbb{P}_i}$, $\ldots$, $M_T^n \in \mathcal{M}^n_{\mathbb{P}_i}$, respectively. For an agent A_i, the realization of $\mathbb{P}^i_1, \mathbb{P}^i_2, \ldots, \mathbb{P}^i_n$ forms the instantiated MDP $M^i_1[u^i_{1_k}], M^i_2[u^i_{2_l}], \ldots, M^i_n[u^i_{n_m}]$ respectively, for some parameter instantiation $u^i_{1_k} \in V_{M^i_1}$, $u^i_{2_l} \in V_{M^i_2}, \ldots, u^i_{n_m} \in V_{M^i_n}$. Therefore, the complete system for A_i for the above instantiations becomes $S[k, l, \ldots, m]^i = M^i_1[u^i_{1_k}] \parallel M^i_2[u^i_{2_l}] \parallel \ldots \parallel M^i_n[u^i_{n_m}] \parallel M^i_T$. The current policy for agent A_i $\sigma_{k,l,\ldots,m} : Path^{S[k,l,\ldots,m]^i}_{fin} \rightarrow Act$ such that it maximizes $P(S[k, l, \ldots, m]^{i^{\sigma}_{k,l,\ldots,m}} \models \varphi)$. For agent A_i, all remaining agents $\mathbb{A} \setminus A_i$ are considered as the environment. All possible policies of agent A_i form a set Σ_a, and all environment policy forms Σ_e.

Problem. Given the true models of n number of agents $A_1, A_2, \ldots, A_n$ as $M_T^1, M_T^2, \ldots, M_T^n$, and the uMDP model of all other agents to A_i as $\mathcal{M}^1_{\mathbb{P}^i_1}, \mathcal{M}^2_{\mathbb{P}^i_2}, \ldots, \mathcal{M}^n_{\mathbb{P}^i_n}$, respectively, and initial policy of A_i is σ^i_{init}, a specification φ and a threshold value ϵ.

- **Question 1.** Does the policy σ^i_{init} satisfies $P(S[k, l, \ldots, m]^{i^{\sigma}_{k,l,\ldots,m}} \models \varphi) \geq \epsilon$ for all realization of $\mathbb{P}^i_1, \mathbb{P}^i_2, \ldots, \mathbb{P}^i_n$ with $M^i_1[u^i_{1_k}], M^i_2[u^i_{2_l}], \ldots, M^i_n[u^i_{n_m}]$ where $u^i_{1_k} \in V_{M^i_1}$, $u^i_{2_l} \in V_{M^i_2}, \ldots, u^i_{n_m} \in V_{M^i_n}$ and $\epsilon \in [0, 1]$?
- **Question 2.** If σ^i_{init} does not work for all realization of $\mathbb{P}^i_1, \mathbb{P}^i_2, \ldots, \mathbb{P}^i_n$, what is the new policy $\sigma_{k,l,\ldots,m}$ such that $P(S[k, l, \ldots, m]^{i^{\sigma_{k,l,\ldots,m}}} \models \varphi) = \arg \underset{\pi \in \Sigma_{A_i}}{max} \underset{\tau \in \Sigma_e}{min} P(S[k, l, \ldots, m]^{\pi,\tau} \models \varphi)$, where Σ_{A_i} is all policy of A_i and Σ_e is the all policy of $\mathbb{A} \setminus A_i$?
- **Question 3.** In case no realistic policy $\sigma_{k,l,\ldots,m}$ exists, how to update the behavior of agent A_i with minimum change in model?

Example. To provide the intuition of the problem statement in a real-world situation, let's consider an autonomous vehicle A_a interacting with another vehicle A_h. The true behavior of these two vehicles is modeled by MDP M_a and M_h, respectively. The specification is $\varphi = (\neg\ crash\ U\ goal)$, where a "crash" happens when both vehicles share the same location and the goal for A_i is s_7. For simplicity, A_a model the behavior of A_h as the uMC $M^a_{\mathbb{P}_h} = (M^a_h, \mathbb{P}_h)$ where pMC $M^a_h = (S_h, s_1, V^h, P^h, AP, L)$, and $\mathbb{P}^h$ is a probability distribution over the parameter space V_{M^h}. For each $u \in V_{M^h}$, a realization of $\mathbb{P}_h$ yield a instantiated MC $M^a_h[u]$. Here, $S_h = \{s_0, \ldots, s_6\}$, $V^h = \{v\}$, AP and L remain same as normal MC. For any two states $s_i, s_j \in S_h$, the $P^h : S_h \times S_h \rightarrow \mathbb{Q}[V^h]$ is defined as

$$P^h(s_i, s_j) = \begin{cases} v, & \text{if j=i+1 and i=\{0,\ldots, 5\}} \\ 1 - v, & \text{if i==j and i=\{0,\ldots, 5\}} \\ 1, & \text{if i==j and i=7} \\ 0, & \text{Otherwise} \end{cases} \tag{1}$$

The true model of A_a is modeled as MDP $M_a = (S_a, s_0, Act, P, AP, L)$ where $S_a = \{s_0, \ldots, s_7\}$, $Act = \{brake, acc\}$, the transition probability $P : S_a \times Act \times S_a \rightarrow [0, 1]$ is defined as follows:

$$P(s_i, acc, s_j) = \begin{cases} q1, & \text{if j=i+1 and i=\{0,..., 5\}} \\ 1-q1, & \text{if i==j and i=\{0,..., 5\}} \\ 1, & \text{if i==j and i=7} \\ 0, & \text{Otherwise} \end{cases} \tag{2}$$

$$P(s_i, brake, s_j) = \begin{cases} q2, & \text{if j=i+1 and i=\{0,..., 5\}} \\ 1-q2, & \text{if i==j and i=\{0,..., 5\}} \\ 1, & \text{if i==j and i=6} \\ 0, & \text{Otherwise} \end{cases} \tag{3}$$

Where $q1, q2 \in [0, 1]$, and depending on the concrete value of q1 and q2, the concrete MDP M_a is formed. Similarly, the true model of A_h is modeled as MDP $M_h = (S_h, s_1, Act1, P1, AP, L)$ where $S_h = \{s_0, \dots, s_6\}$, $Act1 = \{brake, acc\}$, the transition probability $P1 : S_h \times Act1 \times S_h \to [0, 1]$ is defined as follows:

$$P1(s_i, acc, s_j) = \begin{cases} q3, & \text{if j=i+1 and i=\{0,..., 5\}} \\ 1-q3, & \text{if i==j and i=\{0,..., 5\}} \\ 1, & \text{if i==j and i=6} \\ 0, & \text{Otherwise} \end{cases} \tag{4}$$

$$P1(s_i, brake, s_j) = \begin{cases} q4, & \text{if j=i+1 and i=\{0,..., 5\}} \\ 1-q4, & \text{if i==j and i=\{0,..., 5\}} \\ 1, & \text{if i==j and i=6} \\ 0, & \text{Otherwise} \end{cases} \tag{5}$$

Where $q3, q4 \in [0, 1]$, and depending on the concrete value of $q3$ and $q4$, the concrete MDP M_h is formed.

We assumed that the initial policy of A_a from the historical data is $\sigma^a_{init} = \{\sigma(s) : acc, \forall s \in S_a\}$. In the scope of this project, we are not concerned about how σ^a_{init} is coming from, but we plan to use machine learning to predict the initial σ^a_{init} from the available historical data in future.

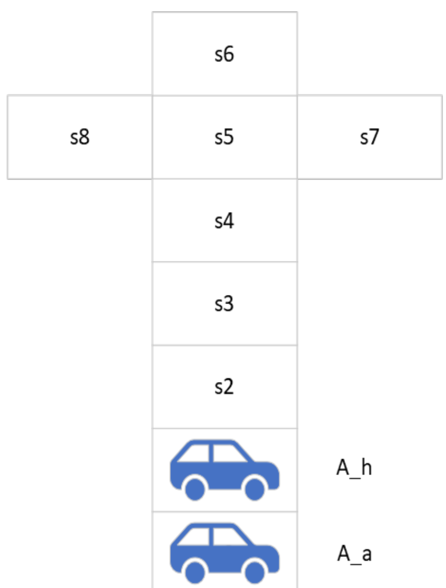

Figure 1: Road and its grid cells in the example scenario along with two vehicles on a straight street. The autonomous vehicle, A_a, originates from position s_0, while vehicle A_h starts from s_1. The goal of A_h is to reach s_6, and A_a's goal is to reach s_7.

4 Approach

In order to perform this experiment, agent A_i constructs the uMC model $M^a_{\mathbb{P}_h}$ of agent A_h. The parameter space of uMC $M^a_{\mathbb{P}_h}$ is V_{M^h}. For each $u \in V_{M^h}$, a realization of $\mathbb{P}_h$ yield an instantiated MC $M^a_h[u]$. Now, A_a builds the complete system $S[u] = M_a \parallel M^a_h[u]$ for the above realization of $\mathbb{P}_h$. We employ the existing TuLIP [8] toolbox to compute the composed system S. More specifically, we used the *synchronous_parallel* function of TuLIP to compute the composed system S. However, considering all possible realizations of $\mathbb{P}_h$ is unrealistic because there exists an infinite number of realizations in the given $\mathbb{P}_h$. To resolve this issue, we generated n real numbers within the interval $[0, 1]$ following function $f : \mathbb{Z} \times \mathbb{Z} \to \mathbb{R}$. For generating $n = 100$ samples within $[0, 1]$, the function is implemented to generate any k^{th} real number as $f(n, k) = \frac{k}{n+1}$. All the finite realization of $\mathbb{P}_h$ is expressed as $\mathbb{P}^{fin}_h = \{f(100, k) : \forall k \in \{1, \ldots, 100\}\}$. Agent A_a can now verify the given specification φ on the complete system S using the provided function *model_checking* in the TuLIP.

In the following, we will explain the steps to answer each question.

Question 1. To answer this question based on the above complete system, it calculates complete system $S[k]$ for each $k \in \mathbb{P}^{fin}_h$. Next, it checks condition $P(S[k]^{\sigma^a_{init}} \models \varphi) \geq \epsilon$. If each $P(S[k]^{\sigma^a_{init}} \models \varphi) \geq \epsilon$ is satisfied for each $k \in \mathbb{P}^{fin}_h$, then agent A_a does not need to update its policy. This implies the computed policy σ^a_{init} works for all possible realizations of $\mathbb{P}^{fin}_h$. In case it finds a single realization for which $P(S[k]^{\sigma^a_{init}} \models \varphi) \geq \epsilon$ is violated, it updates its policy so that new policy σ^a_{new} satisfies $P(S[k]^{\sigma^a_{new}} \models \varphi) \geq \epsilon$ for $k \in \mathbb{P}^{fin}_h$. The second question answers the processor of extracting policy σ^a_{new}.

Question 2. In this example, we do not need to answer this question because we assumed agent A_a considered the uMC model of other agents. Therefore, agents A_h would not have any policy due to having no nondeterminism. As a result, agent A_a could not change the policy to satisfy the specification φ. However, if A_a observes other agent A_h's behavior as uMDP $\mathcal{M}^h_{\mathbb{P}^a_h} = (M^a_h, \mathbb{P}^a_h)$, where pMDP $M^a_h = (S_h, Act, s_{h_0}, V^a_h, P^a_h, AP, L)$, and $\mathbb{P}^a_h$ is a probability distribution over the parameter space $V_{M^a_h}$, then for each $u \in V_{M^a_h}$, the instantiated MDP becomes $M^a_h[u]$, and A_a may change policy to satisfy φ. If Σ^a_h becomes all policy for this instantiated MDP $M^a_h[u]$, and Σ_a becomes all policy for A_a then it finds one policy $\sigma_u \in \Sigma_a$ such that $P(S[u]^{\sigma_a} \models \varphi) = \arg \underset{\pi \in \Sigma_a}{max} \underset{\tau \in \Sigma^a_h}{min} P(S[u]^{\pi,\tau} \models \varphi)$. For finite realizations of $\mathbb{P}^{fin}_h$, the optimal policy is added to a list Σ^{opt}_a. Now, iterate over all policy Σ^{opt}_a and checks if there exists one $\sigma^a_{new} \in \Sigma^{opt}_a$ such that it satisfies $P(S[u]^{\sigma^a_{new}} \models \varphi) \geq \epsilon$ for all realization of its parameter space.

Question 3. This questions is the special case when the initial policy σ_{init} and $\sigma^{opt} \in \Sigma^{opt}_a$ fail to satisfy $P(S[k]^{\sigma_{init}} \models \varphi) \geq \epsilon$ and $P(S[k]^{\sigma^{opt}} \models \varphi) \geq \epsilon$ for $k \in \mathbb{P}_h$. So, A_a changes its own model so that $P(S[k]^{\sigma_{init}} \models \varphi) \geq \epsilon$ is satisfied for all realization of $\mathbb{P}^{fin}_h$. To change its model, it repeatedly updates the model and constructs the complete system S. Then it checks if $P(S[k]^{\sigma^a_{init}} \models \varphi) \geq \epsilon$ is satisfied. This process is repeated until $P(S[k]^{\sigma^a_{init}} \models \varphi) \geq \epsilon$ is met.

5 Experimental results

This section explains the result of the three questions discussed above. We considered the initial policy from historical data as σ^a_{init}, which selects action acc in all states.

Question 1. This question finds an answer if the policy extracted from the historical data works for all realization of $\mathbb{P}^{fin}_h$. It is clear from the Figure 2 that $P(\varphi = \neg\, crash\ U\ goal)$ does not satisfy the condition of being greater than ϵ for all realization of $\mathbb{P}^{fin}_h$. Which means that $P(S[k]^{\sigma^a_{init}} \models \varphi) \geq \epsilon$ is violated at least for some k. Therefore, agents need to find a new policy σ^a_{new} so that $P(S[k]^{\sigma^a_{new}} \models \varphi) \geq \epsilon$ is satisfied for possible realization of $k \in \mathbb{P}^{fin}_h$.

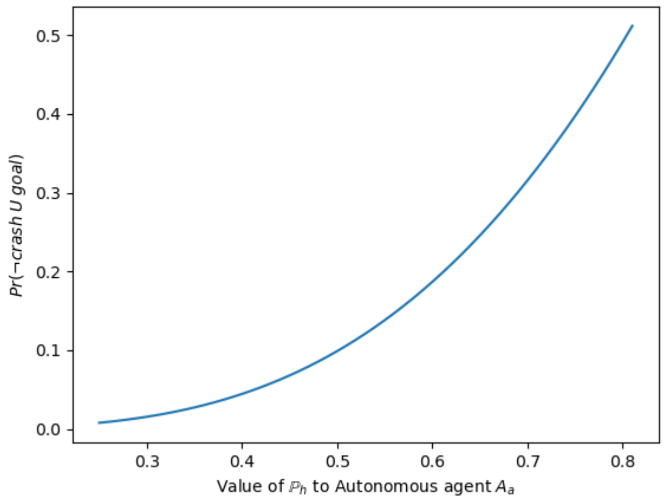

Figure 2: The probability of satisfying the specification, φ for different realization of $\mathbb{P}_h^{fin}$ to A_a. Agent A_a observes $M^a_{\mathbb{P}_h}$, where $v \in [0.25, 0.8]$, and models its own behavior with $q1 = 0.9$ and $q2 = 0.1$, and $\epsilon = 0.20$

Question 2. As A_a considers the uMC model of A_h, no policy σ^a_{new} exists for A_h.

Question 3. As there exists no new policy σ^a_{new} for A_h, agent A_h needs to update its model with minimum change to satisfy $P(S[k]^{\sigma^a_{init}} \models \varphi) \geq \epsilon$ for $\forall k \in \mathbb{P}_h^{fin}$. The experimental result shows that if agent A_a updates its model to the probability of moving to the next state using action 'acc' as q1=.29, then its initial policy satisfies $P(S[k]^{\sigma^a_{init}} \models \varphi) \geq \epsilon$ for $k \in \mathbb{P}_h$.

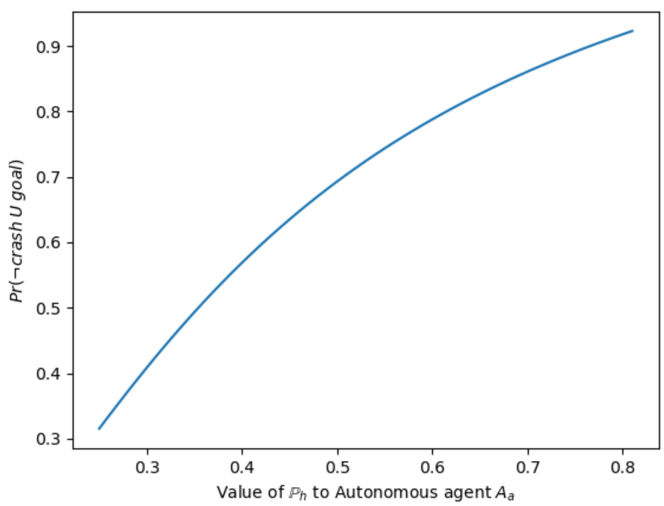

Figure 3: The probability of satisfying the specification, φ for different realization of $\mathbb{P}_h^{fin}$ to A_a. Agent A_a observes $M^a_{\mathbb{P}_h}$, where $v \in [0.25, 0.8]$, and models its own behavior with $q1 = 0.29$ and $q2 = 0.1$ and $\epsilon = 0.20$

6 Conclusion and Future Work

This work evaluates the initial policy and decides whether it satisfies the safety property for all valuations of parameters in a given parameter space using existing formal verification techniques. It also suggests a new policy if the initial policy fails. Additionally, it shows an approach to update the model of an agent if both the initial and new policies cannot work. As a part of future research direction, we will implement a machine learning model to predict the initial policy so that we do not need to rely on the assumption regarding the initial policy. Next, we plan to extend this work so that agents observe the other agent's behavior as the uMDP; currently, it only considers uMC. Finally, we will implement our work in an autonomous vehicle simulator, CARLA, to observe how our overall approach works in a simulated environment.

References

[1] S. Junges, E. Abraham, C. Hensel, N. Jansen, J.-P. Katoen, T. Quatmann, and M. Volk, "Parameter synthesis for markov models," 2019.

[2] M. Cubuktepe, N. Jansen, S. Junges, J.-P. Katoen, and U. Topcu, "Scenario-based verification of uncertain mdps," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2020, pp. 287–305.

[3] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[4] S. Carr, T. Wongpiromsarn, and U. Topcu, "Quantifying faulty assumptions in heterogeneous multi-agent systems," in *2023 7th IEEE Conference on Control Technology and Applications*, 2023, to appear.

[5] G. Bacci, M. Hansen, and K. G. Larsen, "Model checking constrained markov reward models with uncertainties," in *Quantitative Evaluation of Systems: 16th International Conference, QEST 2019, Glasgow, UK, September 10–12, 2019, Proceedings 16*. Springer, 2019, pp. 37–51.

[6] T. Wongpiromsarn, A. Ulusoy, C. Belta, E. Frazzoli, and D. Rus, "Incremental synthesis of control policies for heterogeneous multi-agent systems with linear temporal logic specifications," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 5011–5018.

[7] E. M. Wolff, U. Topcu, and R. M. Murray, "Robust control of uncertain markov decision processes with temporal logic specifications," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 3372–3379.

[8] T. Wongpiromsarn, U. Topcu, N. Ozay, H. Xu, and R. M. Murray, "Tulip: a software toolbox for receding horizon temporal logic planning," in *Proceedings of the 14th international conference on Hybrid systems: computation and control*, 2011, pp. 313–314.