2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Model-Free Reinforcement Learning for Optimal Control of Markov Decision Processes Under Signal Temporal Logic Specifications

Krishna C. Kalagarla, Rahul Jain, Pierluigi Nuzzo
Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles
Email: {kalagarl,rahul.jain,nuzzo}@usc.edu

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9683444

Abstract— **We present a model-free reinforcement learning (RL) algorithm to find an optimal policy for a finite-horizon Markov decision process (MDP) while guaranteeing a desired lower bound on the probability of satisfying a signal temporal logic (STL) specification. We propose a method to effectively augment the MDP state space to capture the required state history and express the STL objective as a reachability objective. The planning problem can then be formulated as a finite-horizon constrained Markov decision process (CMDP). For a general finite-horizon CMDP problem with unknown transition probability, we develop a reinforcement learning scheme that can leverage any model-free RL algorithm to provide an approximately optimal policy out of the general space of non-stationary randomized policies. We illustrate our approach in the context of robotic motion planning for complex missions under uncertainty and performance objectives.**

I. Introduction

Markov decision processes (MDPs) [1] offer a natural framework to express sequential decision-making problems and have increasingly been combined with temporal logic specifications [2] to rigorously express complex mission objectives or constraints. In particular, signal temporal logic (STL) [3] is a rich temporal extension of propositional logic that can express continuous-time continuous-valued signals and can be used, for instance, to unambiguously capture bounds on physical variables or time-sensitive objectives.

Previous efforts have focused on maximizing the probability of satisfying a given STL specification [4]–[6], for example, by maximizing a log-sum-exp approximation of the satisfaction probability. However, in many applications, mission-critical requirements, involving stronger guarantees on the satisfaction of temporal logic objectives, must be paired with performance constraints, such as smoothness of motion, or fuel consumption rates, usually expressed in terms of cost functions [7]. The focus of this paper is on these composite tasks where a total cost on an MDP must be minimized while guaranteeing a lower bound on the probability of satisfying a given STL specification. In particular, we consider a bounded-time fragment of STL that allows up to two layers of nested temporal operators and is expressive enough to capture objectives such as "eventually reach a location within t_1 minutes and remain there for t_2 minutes." To the best of our knowledge, this is the first paper addressing this problem formulation.

Our contribution is twofold. We first propose a method that extends and modifies a previously proposed technique [4] to efficiently augment the state space of the MDP and reduce STL satisfaction to a reachability objective for a finite-horizon MDP. We can then cast the logically-constrained optimal control problem as the problem of controlling a finite-horizon constrained Markov decision process (CMDP) [8]. As in previous approaches [4], [5], we augment the MDP state space to be able to reason about the satisfaction of the STL formula. However, our method allows formulating probabilistic constraints on STL satisfaction and additional cost objectives which could not be expressed within a log-sum-exp formulation.

For a general finite-horizon CMDP problem with unknown transition probability, we further introduce a model-free reinforcement learning (RL) scheme that produces an approximately optimal policy out of the general space of non-stationary randomized policies. Specifically, we formulate the CMDP problem as a min-max game between a player utilizing a no-regret algorithm and a player using a model-free RL algorithm [1], [9]. Our scheme can use any model-free RL algorithm and provides guarantees that the performance of the returned policy can be made arbitrarily close to that of the optimal policy.

A min-max game formulation was also used in the past to find optimal mixed deterministic policies in the context of offline RL for discounted CMDPs [10] as well as feasible policies satisfying a set of convex constraints without optimality guarantees with respect to a cost objective [11]. Differently from these efforts, we focus on finite-horizon CMDPs and use the concept of occupancy measures [8], [12] to obtain an approximately optimal policy out of the general space of non-stationary randomized policies. We illustrate the applicability of our approach on two cases studies, showing that the returned policies very closely satisfy the probabilistic STL constraints and have performance comparable to that of the optimal policies.

II. Preliminaries

We denote the sets of real, non-negative real and natural numbers by $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{N}$, respectively. The indicator function $\mathbb{1}_{s_0}(s)$ evaluates to 1 when $s = s_0$ and 0 otherwise. The probability simplex over the set S is denoted by Δ_S.

Signal Temporal Logic (STL). We use a fragment of signal temporal logic (STL) [3], a temporal extension of propositional logic, to specify complex tasks. The STL formulae in this paper are constructed inductively as follows:

$$\begin{aligned} \Phi_o &:= \mathbf{F}_{[0,T_o]}\Phi_{in} \mid \mathbf{G}_{[0,T_o]}\Phi_{in}, \\ \Phi_{in} &:= \Phi_{in} \wedge \Phi_{in} \mid \Phi_{in} \vee \Phi_{in} \mid \mathbf{F}_{[0,T_{in}]}\varphi \mid \mathbf{G}_{[0,T_{in}]}\varphi, \\ \varphi &:= \mathsf{true} \mid p \mid \neg\varphi \mid \varphi \wedge \varphi, \end{aligned} \tag{1}$$

where $T_o, T_{in} \in \mathbb{R}_+$, Φ_o, Φ_{in}, and φ are STL formulae, and p is a predicate of the form $f(\sigma) < d$, where $\sigma : \mathbb{R}_+ \to \mathbb{R}^n$ is a signal and $f(\sigma) : \mathbb{R}^n \to \mathbb{R}$ is a function mapping a signal value to the real line. Further, $\wedge$ and $\neg$ are the logic conjunction and negation, and $\mathbf{F}$ and $\mathbf{G}$ are the *eventually* and *always* temporal operators.

The Boolean semantics of our STL formulae are interpreted over finite-length signals. Let $\sigma(t)$ be the value of the signal at time t, (σ, t) be the suffix of the signal σ starting from time t, and $\sigma_{t_1:t_2}$ be the segment of the signal from time t_1 to time t_2. Informally, signal (σ, t) satisfies a predicate p, written $(\sigma, t) \models p$, if the signal value at time t, $\sigma(t)$, satisfies p. The signal (σ, t) satisfies $\mathbf{F}_{[a,b]}\phi$ if there exists $a \leq t' \leq b$ such that $(\sigma, t+t')$ satisfies ϕ. Finally, signal (σ, t) satisfies $\mathbf{G}_{[a,b]}\phi$ if $(\sigma, t+t')$ satisfies ϕ for all $a \leq t' \leq b$.

Let $((\sigma, t) \models \phi)$ evaluate to 1 if true and 0 otherwise. Then, we have the following equivalences:

$$\begin{aligned}
\exists\ t' \in [a,b], (\sigma, t') \models \phi &\iff \max_{t' \in [a,b]} ((\sigma, t') \models \phi) = 1, \\
\forall\ t' \in [a,b], (\sigma, t') \models \phi &\iff \min_{t' \in [a,b]} ((\sigma, t') \models \phi) = 1, \\
(\sigma, t) \models \phi_1 \wedge (\sigma, t) \models \phi_2 &\iff \min_{i=1,2} \{(\sigma, t) \models \phi_i\} = 1, \\
(\sigma, t) \models \phi_1 \vee (\sigma, t) \models \phi_2 &\iff \max_{i=1,2} \{(\sigma, t) \models \phi_i\} = 1.
\end{aligned}$$

While allowing for only two layers of nested temporal operators, this STL fragment allows specifying a rich set of time-bounded and safety requirements. The *horizon* $hrz(\phi)$ [3] of an STL formula ϕ is the minimum time length needed to certify whether a signal satisfies the formula or not.

Finite-Horizon MDPs. We consider finite-horizon MDPs [1], which can be formally defined by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, s_0, p, c)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the finite state and action spaces, respectively. The agent interacts with the environment in episodes of length $H+1$, each episode starting with the same initial state s_0. The non-stationary transition probability is denoted by p where $p_h(s'|s,a)$ is the probability of transitioning to state s' upon taking action a at state s at time step $h \in \{0, \ldots, H\}$. The deterministic, non-stationary cost of taking action a in state s at time step h is $c_h(s,a) \in [0, \bar{C}]$.

A non-stationary randomized policy $\pi = (\pi_0, \ldots, \pi_H) \in \Pi$, where $\pi_i : \mathcal{S} \to \Delta_{\mathcal{A}}$, maps each state to a probability simplex over the action space. For a state $s \in \mathcal{S}$ and time step $h \in \{0, \ldots, H\}$ the value function of a non-stationary randomized policy $V_h^\pi(s;c)$ is defined as $V_h^\pi(s;c) = \mathbb{E}\left[\sum_{i=h}^{H} c_i(s_i, a_i) | s_h = s, \pi, p\right]$, where the expectation is over the environment and policy randomness. In the following, we omit π and c when they are clear from the context. The total expected cost of an episode under policy π with respect to cost function c is the respective value function from the initial state s_0, i.e., $V_0^\pi(s_0;c)$. There always exists an optimal non-stationary deterministic policy π^* [1] such that $V_h^{\pi^*}(s) = V_h^*(s) = \inf_\pi V_h^\pi(s)$.

Since the STL formulae are defined over a continuous time as opposed to discrete-step MDPs, we discretize the continuous time space by considering a step size Δt. Without loss of generality, we take $\Delta t = 1$. A finite *run* ξ_t of the MDP at time $t \in \mathbb{N}$ is a sequence of states and actions $s_0 a_0 s_1, a_1 \ldots s_t$ up to time t. Given an MDP $\mathcal{M}$ and an STL formula Φ, a finite run $\xi_t = s_0 a_0 \ldots s_t$, $t \geq hrz(\Phi)$, of the MDP under policy π is said to satisfy Φ if the signal $s_{0:t} = s_0 s_1 \ldots s_t$ generated by the run satisfies Φ. The probability that a run of $\mathcal{M}$ satisfies Φ under policy π is denoted by $Pr_{\mathcal{M}}^\pi(\Phi)$, i.e., $Pr_{\mathcal{M}}^\pi(\Phi) = Pr_{\mathcal{M}}^\pi(s_{0:hrz(\Phi)} \models \Phi)$.

Finite-Horizon Constrained MDPs. A finite-horizon constrained MDP (CMDP) [8] is a finite-horizon MDP with an additional constraint expressed by a pair of cost function and threshold $\{d, l\}$. For simplicity, in this paper, we consider a single constraint. Extensions to the case of multiple constraints are straightforward. The cost of taking action a in state s at time step $h \in \{0, \ldots, H\}$ with respect to the constraint cost function is $d_h(s,a) \in [0, \bar{D}]$.

Solving a CMDP problem consists in finding a policy which minimizes the total expected objective cost such that the total expected constraint cost is less than or equal to its threshold l. Formally,

$$\begin{aligned}
\pi^* \in \ \underset{\pi \in \Pi}{\operatorname{argmin}} \quad & V_0^\pi(s_0; c) \\
\text{s.t.} \quad & V_0^\pi(s_0; d) \leq l.
\end{aligned} \tag{2}$$

The optimal value is $V^* = V_0^{\pi^*}(s_0;c)$. The optimal policy may be randomized [8], i.e., an optimal deterministic policy may not exist as in the case of finite-horizon MDPs.

Occupancy Measures. Occupancy measures [8] allow for an alternative representation of the set of non-stationary randomized policies and a formulation of the optimization problem (2) as a linear program (LP). The occupancy measure q^π of a policy π in a finite-horizon MDP is defined as the expected number of visits to a state-action pair (s,a) in an episode at time step h. Formally, $q_h^\pi(s,a) = Pr\left[s_h = s, a_h = a | s_0 = s_0, \pi\right]$.

The occupancy measure q^π of a policy π satisfies linear constraints [8] expressing non-negativity and conservation of probability flow through the states. The space of the occupancy measures satisfying these constraints is denoted by $\Delta(\mathcal{M})$ and is convex. A policy π generates an occupancy measure $q \in \Delta(\mathcal{M})$ if

$$\pi_h(a|s) = \frac{q_h(s,a)}{\sum_b q_h(s,b)}, \quad \forall (s,a,h). \tag{3}$$

Thus, there exists a generating policy for all occupancy measures in $\Delta(\mathcal{M})$ and *vice versa*. Further, the total expected cost of an episode under policy π with respect to cost function c can be expressed in terms of the occupancy measure as $V_0^\pi(s_0;c) = \sum_{h,s,a} q_h^\pi(s,a) c_h(s,a)$.

III. Problem Formulation

For a given finite-horizon MDP and STL specification, we are interested in finding a policy which minimizes the total expected cost such that the probability of satisfying the given STL specification is above a given threshold. We assume that the MDP horizon exceeds by one step the horizon of the STL specification. Our formulation can be trivially extended to longer MDP horizons. We then define the following problem.

Problem 1. Given the MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, s_0, p, c)$, the STL formula Φ_o with horizon $H = hrz(\Phi_o) + 1$, and the satisfaction threshold p_{thres}, find a policy π^* such that

$$\begin{aligned}
\pi^* \in \ \underset{\pi \in \Pi}{\operatorname{argmin}} \quad & \mathbb{E}\left[\sum_{i=0}^{H} c_i(s_i, a_i) | s_0 = s_0, \pi\right] \\
\text{s.t.} \quad & Pr_{\mathcal{M}}^\pi(\Phi_o) \geq p_{thres},
\end{aligned} \tag{4}$$

where $Pr^{\pi}_{\mathcal{M}}(\Phi_o)$ is the probability of satisfying Φ_o under policy π.

Because the objective in (4) is not additive in nature and the dependence on the history for determining the probability of satisfying the STL formula is non-Markovian, we need to extend the state space of the MDP to capture the necessary history and evaluate the satisfaction of the formula. In the extended state space, we show that the probability of satisfaction is equal to the probability of reaching a set of states, which can be expressed by a cost function on the extended MDP. The cost function c of the original MDP can also be trivially extended, leading to a standard finite-horizon CMDP formulation. We detail this reduction in Section IV. In Section V, we introduce a model-free reinforcement learning (RL) algorithm to find an ϵ-optimal policy for a given finite-horizon CMDP. This algorithm is then applied to the CMDP resulting from our original problem.

IV. Reduction to CMDP

The STL formula Φ_o is of the form $\mathbf{F}_{[0,T_o]}\Phi_{in}$ or $\mathbf{G}_{[0,T_o]}\Phi_{in}$. Let Φ_{in} include n sub-formulae Φ^i_{in} of the form $\mathbf{F}_{[0,T_{in}]}\varphi^i$ or $\mathbf{G}_{[0,T_{in}]}\varphi^i$, $i = 1,\dots,n$. Each of these sub-formulae has horizon $hrz(\Phi^i_{in}) = T_{in}, \forall\, i, n$. Therefore, the horizon of Φ_{in} is also equal to T_{in}, while the one of Φ_o is $\tilde{H} = T_{in} + T_o$. We then obtain

$$
\begin{aligned}
&s_{0:\tilde{H}} \models \Phi_o \\
\iff &\begin{cases} \max\limits_{t\in[0,T_o]} ((s,t) \models \Phi_{in}) = 1, \Phi_o = \mathbf{F}_{[0,T_o]}\Phi_{in}, \\ \min\limits_{t\in[0,T_o]} ((s,t) \models \Phi_{in}) = 1, \Phi_o = \mathbf{G}_{[0,T_o]}\Phi_{in} \end{cases}, \\
\iff &\begin{cases} \max\limits_{t\in[T_{in},\tilde{H}]} (s_{t-T_{in}:t} \models \Phi_{in}) = 1, \Phi_o = \mathbf{F}_{[0,T_o]}\Phi_{in}, \\ \min\limits_{t\in[T_{in},\tilde{H}]} (s_{t-T_{in}:t} \models \Phi_{in}) = 1, \Phi_o = \mathbf{G}_{[0,T_o]}\Phi_{in}, \end{cases} \\
\iff &\begin{cases} \max\limits_{t\in[T_{in},\tilde{H}]} Sat(s_{t+1},\Phi_{in}), \Phi_o = \mathbf{F}_{[0,T_o]}\Phi_{in}, \\ \min\limits_{t\in[T_{in},\tilde{H}]} Sat(s_{t+1},\Phi_{in}), \Phi_o = \mathbf{G}_{[0,T_o]}\Phi_{in}. \end{cases}
\end{aligned} \tag{5}
$$

where $Sat(s_t, \Phi_{in})$ evaluates to 1 if the signal segment $s_{t-T_{in}-1:t-1}$, i.e., the previous $T_{in}+1$ steps of the signal at time step t satisfies Φ_{in}, and evaluates to 0 otherwise.

We introduce a flag variable fin which, at time step $t+1$, is equal to $\min_{k\in[T_{in},t]} Sat(s_{k+1},\Phi_{in})$ for $\Phi_o = \mathbf{G}_{[0,T_o]}\Phi_{in}$ and equal to $\max_{k\in[T_{in},t]} Sat(s_{k+1},\Phi_{in})$ for $\Phi_o = \mathbf{F}_{[0,T_o]}\Phi_{in}$. This flag fin takes values in the set $FIN = \{0, 1, \perp\}$. We introduce the placeholder $\perp$ since $Sat(s_{t'}, \Phi_{in})$ is undefined for $t' \le T_{in}$. We similarly define $Sat(s_t, \Phi^i_{in})$, $i = 1,\dots,n$, which evaluates to 1 if the signal segment $s_{t-T_{in}-1:t-1}$ satisfies the STL formula Φ^i_{in} and 0 otherwise. By the syntax in (1) and the assumption that all sub-formulae have the same horizon, we can determine $Sat(s_t, \Phi_{in})$ recursively as:

$$
\begin{aligned}
&Sat(s_{t'}, \Phi^i_{in} \vee \Phi^j_{in}) = \max(Sat(s_t,\Phi^i_{in}), Sat(s_t,\Phi^j_{in})), \\
&Sat(s_{t'}, \Phi^i_{in} \wedge \Phi^j_{in}) = \min(Sat(s_t,\Phi^i_{in}), Sat(s_t,\Phi^j_{in})).
\end{aligned} \tag{6}
$$

We further associate a flag f^i which takes values in the set $F^i = \{0, 1, \dots, T_{in}\}$ with each sub-formulae Φ^i_{in}. These flags are used to evaluate $Sat(s_t, \Phi^i_{in})$ and are updated according to the following function

$$
f^i_{t+1} = \begin{cases} T_{in}+1, & \text{if } s(t) \models \varphi^i,\ \Phi^i_{in} = \mathbf{F}_{[0,T_{in}]}\varphi^i, \\ \max(f^i_t - 1, 0), & \text{if } s(t) \not\models \varphi^i,\ \Phi^i_{in} = \mathbf{F}_{[0,T_{in}]}\varphi^i, \\ \min(f^i_t, T_{in}) + 1, & \text{if } s(t) \models \varphi^i,\ \Phi^i_{in} = \mathbf{G}_{[0,T_{in}]}\varphi^i, \\ 0, & \text{if } s(t) \not\models \varphi^i,\ \Phi^i_{in} = \mathbf{G}_{[0,T_{in}]}\varphi^i. \end{cases} \tag{7}
$$

By the definitions of $\mathbf{G}$ and $\mathbf{F}$, $Sat(s_t, \Phi^i_{in})$ can be evaluated from f^i_t as follows

$$
Sat(s_t, \Phi^i_{in}) = \begin{cases} 1, & \text{if } f^i_t > 0,\ \Phi^i_{in} = \mathbf{F}_{[0,T_{in}]}\varphi^i, \\ 0, & \text{if } f^i_t = 0,\ \Phi^i_{in} = \mathbf{F}_{[0,T_{in}]}\varphi^i, \\ 1, & \text{if } f^i_t = T_{in}+1,\ \Phi^i_{in} = \mathbf{G}_{[0,T_{in}]}\varphi^i, \\ 0, & \text{if } f^i_t < T_{in}+1,\ \Phi^i_{in} = \mathbf{G}_{[0,T_{in}]}\varphi^i. \end{cases} \tag{8}
$$

By the definition of fin we also obtain its update rule

$$
fin_{t+1} = \begin{cases} \perp, & t < T_{in}, \\ Sat(s_{t+1}, \Phi_{in}), & t = T_{in}, \\ \min(Sat(s_{t+1},\Phi_{in}), fin_t), & t > T_{in},\ \Phi_o = \mathbf{G}_{[0,T_o]}\Phi_{in}, \\ \max(Sat(s_{t+1},\Phi_{in}), fin_t), & t > T_{in},\ \Phi_o = \mathbf{F}_{[0,T_o]}\Phi_{in}. \end{cases} \tag{9}
$$

By the definition of the flag variables above, we obtain from (5) that $s_{0:\tilde{H}} \models \Phi_o$ if and only if $fin_{\tilde{H}+1} = 1$, that is, $s_{0:\tilde{H}}$ satisfies Φ_o if and only if the flag variable fin is equal to 1 at time $\tilde{H}+1$, where $\tilde{H} = hrz(\Phi_o)$. The satisfaction of the specification has then been reduced to a reachability condition.

We define a flag-augmented MDP $\mathcal{M}^\times = (\mathcal{S}^\times, \mathcal{A}^\times, H^\times, s^\times_0, p^\times, d^\times, c^\times)$, where $\mathcal{S}^\times = (\mathcal{S} \times F^1 \times \dots \times F^n \times FIN)$, with $s^\times = (s, f^1, \dots, f^n, fin)$, $\mathcal{A}^\times = \mathcal{A}$, $s^\times_0 = (s_0, \underbrace{0,\dots,0}_{n \text{ times}}, \perp)$, and $H^\times = hrz(\Phi_o) + 1$. For the transition probability function $p^\times$, the s component of $s^\times$ is updated according to the original probability transition function p, while the flag variables are updated according to (6)-(9). The cost function $d^\times$ is defined such that the expected cost with respect to $d^\times$ is the probability of reaching states with flag variable fin equal to 1 at time $H^\times$. Thus,

$$
d^\times_h(s, f^1, \dots, f^n, fin, a) = \begin{cases} 1, & \text{if } h = H^\times \text{ and } fin = 1, \\ 0, & \text{otherwise.} \end{cases}
$$

Similarly, the objective cost function c of the original MDP can be extended to the augmented MDP $\mathcal{M}^\times$ as follows:

$$
c^\times_h(s, f^1, \dots, f^n, fin, a) = c_h(s, a).
$$

By the derivations above, we can state the following result.

Theorem 1. *For given MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, s_0, p, c)$, STL formula Φ_o, and desired satisfaction threshold p_{thres}, Problem 1 reduces to the following CMDP problem for the extended MDP $\mathcal{M}^\times$.*

$$
\begin{aligned}
\pi^* \in \ &\underset{\pi\in\Pi^\times}{argmin} \quad V^\pi_0(s^\times_0; c^\times) \\
&\quad s.t. \quad V^\pi_0(s^\times_0; d^\times) \ge p_{thres}.
\end{aligned} \tag{10}
$$

V. The CMDP Learning Problem

We consider the setting where an agent repeatedly interacts with a finite-horizon CMDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, s_0, p, c, \{d, l\})$ in episodes of fixed length H, starting from the same initial state s_0. We assume that the cost functions c, d are known to the learning agent, but the transition probability p is unknown. The main objective is to design a model-free online learning algorithm returning an ϵ-optimal policy. A policy π is said to be ϵ-optimal if the total expected objective cost of an episode under policy π is within ϵ of the optimal value, i.e., $V_0^\pi(s_0; c) \leq V^* + \epsilon$, and the constraints are satisfied within an ϵ tolerance, i.e., $V_0^\pi(s_0; d) \leq l + \epsilon$. We make the following assumption of feasibility, which can be verified for our problem by computing the maximum STL satisfaction probability [4], [5].

Assumption 1. *The given CMDP $\mathcal{M}$ is feasible, i.e., there exists a policy π such that the constraints are satisfied.*

The optimization problem (2) can be formulated in terms of occupancy measures as:

$$q^* \in \underset{q \in \Delta(\mathcal{M})}{\operatorname{argmin}} \quad C(q) \qquad \text{s.t.} \quad D(q) \leq l, \tag{11}$$

where $C(q) = \sum_{h,s,a} q_h(s,a) c_h(s,a)$ and $D(q) = \sum_{h,s,a} q_h(s,a) d_h(s,a)$.

The Lagrangian of this optimization problem is $L(q, \lambda) = C(q) + \lambda(D(q) - l)$, where $\lambda \in \mathbb{R}_+$ is the Lagrangian multiplier. Following standard results from optimization theory [13], the optimization problem (11) can be formulated as the following min-max problem: $\min_{q \in \Delta(\mathcal{M})} \max_{\lambda \in \mathbb{R}_+} L(q, \lambda)$. Further, the functions $C(q)$ and $D(q)$ are linear in q and the set of occupancy measures $\Delta(\mathcal{M})$ expressed by linear constraints is convex. Therefore, by strong duality [13], the optimization problem (2) is also equivalent to the max-min problem $\max_{\lambda \in \mathbb{R}_+} \min_{q \in \Delta(\mathcal{M})} L(q, \lambda)$.

The latter problem can be viewed as a zero-sum game between a λ-player, who seeks to maximize $L(q, \lambda)$, and a q-player, who seeks to minimize $L(q, \lambda)$. We use a previously proposed approach [14] for solving such a game. In this approach, the λ-player plays a no-regret online learning algorithm [15] against the *best response* strategy played by the q-player. In no-regret online learning, the difference between the cumulative gain of the player and that of the best fixed decision in hindsight is sub-linear in the number of plays or iterations. Specifically, for each t, given λ_t played by the λ-player, the q-player plays the *best response* q_t with respect to the loss function $L(q, \lambda_t)$. The λ-player then observes the gain function $l_t(\lambda)$, which is the Lagrangian $L(q_t, \lambda) = C(q_t) + \lambda(D(q_t) - l)$. With this feedback, the λ-player updates the Lagrange multiplier λ according to a no-regret online learning algorithm. We refer to the extended version of this paper [16] for further details.

The *best response* above is the occupancy measure which minimizes the current Lagrangian $L(q, \lambda_t)$, i.e.,

$$\underset{q \in \Delta(\mathcal{M})}{\operatorname{argmin}} L(q, \lambda_t) = \underset{q \in \Delta(\mathcal{M})}{\operatorname{argmin}} C(q) + \lambda_t(D(q) - l).$$

This best response can be calculated by finding the optimal policy of the MDP with respect to cost function $c + \lambda_t d$. The optimal policy is then translated into its associated occupancy measure which is the desired best response.

Algorithm 1 `Meta-Algorithm`

```
Initialize λ_1
for t = 1, ..., T do
    q_t ← Best-Response(λ_t),
    λ-player is given the gain function L(q_t, λ),
    λ_{t+1} ← OnlineLearning(λ_1, q_1, ..., λ_t, q_t).
Return (1/T) Σ_{t=1}^{t=T} q_t.
```

A. Occupancy-Based Model-Free Constrained Reinforcement Learning (`OB-MFC`) Algorithm

We summarize the above approach in Algorithm 1. The `Best-Response` function is implemented in two steps. First, we use a model-free RL algorithm [1], [9] to find an optimal policy with respect to a scalar cost function $c + \lambda_t d$. To ensure finite completion time for the RL algorithm, we can make the simple assumption that the RL algorithm `Best-Response-Policy` returns an ϵ-optimal policy.

Assumption 2. *Given cost functions c, d and $\lambda \in \mathbb{R}_+$, the RL algorithm `Best-Response-Policy` returns a policy π such that $V(\pi) < \min_{\pi' \in \Pi} V(\pi') + \epsilon_{br}$, where $V(\pi)$ is the total expected return with respect to cost function $c + \lambda d$.*

As a second step, the corresponding occupancy measure q_t of policy π_t, which is the desired output of the `Best-Response` function, is estimated by Monte Carlo estimation following the definition of an occupancy measure, i.e., $q_h^\pi(s,a) = Pr\left[s_h = s, a_h = a | s_0 = s_0, \pi\right]$. We make a further assumption that an `Occupancy-Estimator` returns a good estimate of the occupancy measure. This can be ensured by a sufficiently large number of roll-outs for accurate Monte Carlo estimation.

Assumption 3. *Given policy π, `Occupancy-Estimator` returns an occupancy measure estimate $\hat{q}$ such that $\|q - \hat{q}\|_1 \leq \epsilon_{oe}$, where q is the occupancy measure of policy π.*

Most online convex optimization algorithms [15] make a decision from a bounded convex space. We thus require $\lambda \leq B$, where B is a hyper-parameter to be chosen. The scalar λ is then augmented by one more dimension corresponding to $B - \lambda$ to give a bidimensional vector $(\lambda[1], \lambda[2])$. The cost function d can also be seen as being augmented by 0. The online learning agent then chooses λ such that $||\lambda||_1 = B$.

We use the Exponentiated Gradient (EG) [17] online learning algorithm, which is known to be a no-regret algorithm, to implement the `OnlineLearning` function of Algorithm 1. This algorithm utilizes the sub-gradient ∂l_t of the revealed gain function $l_t(\lambda)$, namely, $L(\hat{q}, \lambda) = C(\hat{q}_t) + \lambda[1](D(\hat{q}_t) - l)$, which is nothing but $\left[(D(\hat{q}_t) - l), 0\right]^T$. We denote by $\hat{q}_t$ the estimate of q_t, the occupancy measure associated with π_t, obtained by `Occupancy-Estimator`. This estimate is used to approximate the sub-gradient ∂l_t by using $D(\hat{q}_t) = \sum_{h,s,a} \hat{q}_h(s,a) d_h(s,a)$. By putting all this together, we obtain the occupancy-based model-free constrained reinforcement learning (`OB-MFC`) Algorithm 2.

Algorithm 2 `OB-MFC Reinforcement Learning`

Input: Bound B, learning rate η, number of roll-outs N, number of iterations T.
Initialize $\lambda = (\frac{B}{2}, \frac{B}{2})$.
for $t = 1, \ldots, T$ **do**
  $\pi_t \leftarrow$ `Best-Response-Policy`(λ_t),
  $\hat{q}_t \leftarrow$ `Occupancy-Estimator`(π, N),
  $D(\hat{q}_t) \leftarrow \sum_{h,s,a} \hat{q}_h(s,a) d_h(s,a)$,
  $\beta_t = [(D(\hat{q}_t) - l), 0]^T$,
  $\lambda_{t+1}[i] \leftarrow B \frac{\lambda_t[i] e^{\eta \beta_t[i]}}{\sum_j \lambda_t[j] e^{\eta \beta_t[j]}}$ for $i = 1, 2$.
$\tilde{q} \leftarrow \frac{1}{T} \sum_{t=1}^{T} \hat{q}_t$,
$\tilde{\pi}_h(a|s) \leftarrow \frac{\tilde{q}_h(s,a)}{\sum_b \tilde{q}_h(s,b)}, \quad \forall (s,a,h)$.
Return $\tilde{\pi}$.

B. Optimality of OB-MFC RL Algorithm

In this section, we provide guarantees that the performance of the returned policy with respect to the given CMDP problem can be arbitrarily close to that of the optimal policy. The proofs of the following results can be found in the extended version of the paper [16]. We first show that the difference between the Lagrangian functions $L(q, \lambda)$ and $L(\hat{q}, \lambda)$ with respect to the true occupancy measure q and the estimated occupancy measure $\hat{q}$ is small.

Lemma 1. *Let q be the occupancy measure associated with a policy π and $\hat{q}$ be its empirical estimate such that the L_1 estimation error is small, i.e., $\|q - \hat{q}\|_1 \leq \epsilon_{oe}$. Then, $|L(\hat{q}, \lambda) - L(q, \lambda)| \leq \epsilon_{est}$ for all $||\lambda||_1 = B$, where $\epsilon_{est} = (\bar{C} + B\bar{D})\epsilon_{oe}$.*

We next show that the primal-dual gap falls below a desired threshold ϵ_{ol} after a suitably large number of iterations T of the algorithm.

Lemma 2. *After T iterations of the algorithm, we have*

$$\max_{\lambda \in \mathbb{R}_+^2, ||\lambda||_1 = B} L(\tilde{q}, \lambda) - L(\bar{q}, \tilde{\lambda}) \leq \epsilon_{ol}/2,$$

$$\min_{q \in \Delta(\mathcal{M})} L(q, \tilde{\lambda}) - L(\bar{q}, \tilde{\lambda}) \geq \epsilon_{ol}/2.$$

Thus, the following holds for the primal-dual gap

$$\max_{\lambda \in \mathbb{R}_+^2, ||\lambda||_1 = B} L(\tilde{q}, \lambda) - \min_{q \in \Delta(\mathcal{M})} L(q, \tilde{\lambda}) \leq \epsilon_{ol},$$

where $\tilde{q} = \frac{1}{T} \sum_{t=1}^{T} \hat{q}_t$, $\bar{q} = \frac{1}{T} \sum_{t=1}^{T} q_t$, $\tilde{\lambda} = \frac{1}{T} \sum_{t=1}^{T} \lambda_t$, and $\epsilon_{ol} = 2\epsilon_{br} + 2\epsilon_{est} + \frac{o(T)}{T}$.

We can thus make the primal dual gap arbitrarily small by reducing ϵ_{br} and ϵ_{est} and increasing the number of iterations T. Let the number of iterations T be large enough such that $\frac{o(T)}{T} < \epsilon_{reg}$. Then, we obtain $\epsilon_{ol} < \epsilon_{reg} + 2\epsilon_{br} + \epsilon_{est}$. We now show that the returned average occupancy measure approximately satisfies the constraint and has an expected return close to that of the optimal policy.

Lemma 3. *Under Assumption 1, the returned occupancy measure estimate $\tilde{q}$ approximately satisfies the given constraint*

$$D(\tilde{q}) \leq l + \frac{2(\bar{C}(H+1) + \epsilon_{ol} + \epsilon_{est})}{B}.$$

Further, the objective value returned by $\tilde{q}$ is close to that of the optimal policy

$$C(\tilde{q}) \leq C(q^*) + \epsilon_{reg} + \epsilon_{br} + \epsilon_{est}.$$

The returned $\tilde{q}$ is an estimate of the desired occupancy measure $\bar{q}$. Thus, $\tilde{q}$ may not be a valid occupancy measure, i.e., it may not correspond to a valid policy. Nevertheless, we show that the occupancy measure associated with the policy $\tilde{\pi}$ generated from $\tilde{q}$ is close to $\bar{q}$.

Lemma 4. *Let $\bar{q}$ be the occupancy measure associated with a policy π and $\tilde{q}$ be its empirical estimate such that the L_1 estimation error is small, i.e., $\|\bar{q} - \tilde{q}\|_1 \leq \epsilon_{oe}$. Then, for a policy defined as $\tilde{\pi}_h(a|s) = \frac{\tilde{q}_h(s,a)}{\sum_b \tilde{q}_h(s,b)}, \forall (s,a,h)$, the L_1 error between its associated occupancy measure $\tilde{\tilde{q}}$ and $\bar{q}$ is also small, i.e., $\|\tilde{\tilde{q}} - \bar{q}\|_1 \leq 2(H+1)\epsilon_{oe}$. Furthermore, $\|\tilde{\tilde{q}} - \tilde{q}\|_1 \leq (2H+3)\epsilon_{oe}$ holds.*

From Lemmas 3 and 4, we have:

Theorem 2. *Under Assumption 1, the returned policy $\tilde{\pi}$ approximately satisfies the given constraint*

$$D(\tilde{\pi}) \leq l + \bar{D}(2H+1)\epsilon_{oe} + \frac{2(\bar{C}(H+1) + \epsilon_{ol} + \epsilon_{est})}{B}.$$

Further, the expected objective cost under $\tilde{\pi}$ is close to that of the optimal policy, i.e.,

$$C(\tilde{\pi}) \leq C(\pi^*) + \bar{C}(2H+3)\epsilon_{oe} + \epsilon_{reg} + \epsilon_{br} + \epsilon_{est}.$$

The above result shows that the performance of the returned policy can be made arbitrarily close to that of the optimal policy by making the errors $\epsilon_{ol}, \epsilon_{est}, \epsilon_{reg}$ arbitrarily small and the Lagrange multiplier bound B suitably large. We can attain arbitrarily small $\epsilon_{ol}, \epsilon_{est}$, and ϵ_{reg} by using a sufficiently large number of iterations T, a better `Occupancy-Estimator`, and by running the model-free RL algorithm longer to obtain a policy closer to the optimal best response.

VI. Experimental Results

We implemented our framework in Python and used the LP solver provided by Gurobi to find the optimal cost of a CMDP with a known transition probability. We evaluate our framework on two case studies involving motion planning of a mobile robot. The experiments are run on a 1.4-GHz Core i5 processor with 16-GB memory.

We consider a robot moving with discrete actions in a simple grid world with discrete states as shown in Fig. 1. The set of actions available to the robot in each state is $\mathcal{A} = (\mathsf{N}, \mathsf{E}, \mathsf{S}, \mathsf{W}, \mathsf{NE}, \mathsf{NW}, \mathsf{SE}, \mathsf{SW}, \mathsf{rest})$. The dynamics of the robots is as follows. The action rest does not change the robot state. Also, if the robot cannot move in the intended direction, then it remains in the same state. For all other actions, the robot moves in the intended direction with probability $p = 0.93$ and the remaining probability is equally divided between the following choices: the two possible adjoining directions and staying in the same state, as shown in Fig. 1. For all time steps and states, the cost of action rest is 0, the cost of the horizontal or vertical actions, i.e., $(\mathsf{N}, \mathsf{E}, \mathsf{S}, \mathsf{W})$ is 1, and the cost of the diagonal actions, i.e., $\mathsf{NE}, \mathsf{NW}, \mathsf{SE}, \mathsf{SW}$ is 2. We use the standard model-free Q-learning [1] algorithm to implement `Best-Response-Policy` and

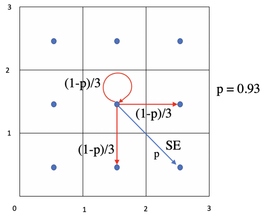

Fig. 1. Robot dynamics for action SE.

Monte-Carlo estimation with $5,000$ trajectories to implement `Occupancy-Estimator`.

A. Case Study 1: Bounded-Time Reachability

In this case study, we consider a grid world of size (6×6) with the robot starting at $(0.5, 0.5)$. The STL formula $\Phi_o = \mathbf{F}_{[0,7]} \mathbf{G}_{[0,1]}(x > 4 \wedge y > 4)$ expresses a requirement of the form "Eventually visit and remain for t_1 units of time in the desired region within t_2 units of time." The horizon of the MDP problem is $hrz(\Phi_o) + 1 = 9$.

We construct the extended MDP as described in Section IV, resulting in an extended state space $\mathcal{S}^\times$ with $|\mathcal{S}^\times| = 324$, and consider two different thresholds for STL satisfaction p_{thres}, i.e., 0.5 and 0.9. Since the transition probability is known by construction in both these cases, an optimal policy and the true optimal total cost is obtained by solving the LP formulation of the finite-horizon CMDP as described in Section IV. The optimal cost for $p_{thres} = 0.5$ and $p_{thres} = 0.9$ is 5.881 and 7.494, respectively.

In the setting of unknown transition probability, an optimal policy is obtained by using the model-free `OB-MFC` algorithm. The resulting policies are used to generate $10,000$ trajectories, and the satisfaction probabilities and expected total costs are estimated. The estimated satisfaction probability for $p_{thres} = 0.5$ and $p_{thres} = 0.9$ is 0.501 and 0.897, respectively. The estimated total expected cost for $p_{thres} = 0.5$ and $p_{thres} = 0.9$ is 6.284 and 7.589, respectively. In both cases, the estimated satisfaction probability and total expected cost of the returned policy is within a small, 6.8% tolerance from the optimal value and satisfaction threshold.

B. Case Study 2: Bounded Time Patrolling

In this case study, we consider a grid world of size (4×4) with the robot starting at $(1.5, 1.5)$. The STL formula $\Phi_o = \mathbf{G}_{[0,12]}\big(\mathbf{F}_{[0,2]}(x > 1 \wedge x < 2 \wedge y > 3 \wedge y < 4) \wedge \mathbf{F}_{[0,2]}(x > 2 \wedge x < 3 \wedge y > 2 \wedge y < 3)\big)$ expresses a requirement of the form "For all time $t \in [0, t_1]$, eventually visit region A in interval $[t, t+h]$ and eventually visit region B in interval $[t, t+h]$." The horizon of the MDP problem is $hrz(\Phi_o) + 1 = 15$.

Similarly to the first case study, we construct an extended state space $\mathcal{S}^\times$ with $|\mathcal{S}^\times| = 768$ and consider a threshold for STL satisfaction $p_{thres} = 0.7$. For known transition probability, an optimal policy and the true optimal total cost are obtained by solving the LP formulation of the finite-horizon CMDP as described in Section IV. The optimal cost for $p_{thres} = 0.7$ is 16.875.

In the setting of unknown transition probability, an optimal policy is obtained by using the `OB-MFC` algorithm. The resulting policy is used to generate $10,000$ trajectories and the satisfaction probability and expected total cost are estimated. The estimated satisfaction probability for $p_{thres} = 0.7$ is 0.702 and the estimated total expected cost is 17.215. The estimated satisfaction probability of the returned policy satisfies the given threshold and the estimated total expected cost is within a small, 2.01% tolerance from the optimal value.

VII. Conclusions

We designed and validated a model-free reinforcement learning algorithm for a general finite-horizon constrained Markov decision process and applied it to find a cost-optimal policy for a finite-horizon Markov decision process such that the probability of satisfying a given signal temporal logic (STL) specification is beyond a desired threshold. Future plans include the extension of the proposed method to more general STL specifications and the optimization of the robust satisfaction of STL formulae.

Acknowledgments

This research was supported in part by the US National Science Foundation (NSF) under Awards 1839842 and 1846524, the Office of Naval Research (ONR) under Award N00014-20-1-2258, and the Defense Advanced Research Projects Agency (DARPA) under Award HR00112010003.

References

[1] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1994.

[2] C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT Press, 2008.

[3] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*. Springer, 2004, pp. 152–166.

[4] H. Venkataraman, D. Aksaray, and P. Seiler, "Tractable reinforcement learning of signal temporal logic objectives," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 308–317.

[5] D. Aksaray, A. Jones, Z. Kong, M. Schwager, and C. Belta, "Q-learning for robust satisfaction of signal temporal logic specifications," in *IEEE Conference on Decision and Control (CDC)*, 2016, pp. 6565–6570.

[6] P. Varnai and D. V. Dimarogonas, "On robustness metrics for learning STL tasks," in *American Control Conference (ACC)*. IEEE, 2020, pp. 5394–5399.

[7] K. C. Kalagarla, R. Jain, and P. Nuzzo, "Optimal control of discounted-reward Markov decision processes under linear temporal logic specifications," in *Proc. American Control Conference*, 2021, pp. 1268–1274.

[8] E. Altman, *Constrained Markov Decision Processes*. CRC Press, 1999, vol. 7.

[9] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan, "Is Q-learning provably efficient?" *arXiv preprint arXiv:1807.03765*, 2018.

[10] H. Le, C. Voloshin, and Y. Yue, "Batch policy learning under constraints," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3703–3712.

[11] S. Miryoosefi, K. Brantley, H. Daumé III, M. Dudík, and R. Schapire, "Reinforcement learning with convex constraints," *arXiv preprint arXiv:1906.09323*, 2019.

[12] K. C. Kalagarla, R. Jain, and P. Nuzzo, "A Sample-Efficient Algorithm for Episodic Finite-Horizon MDP with Constraints," in *Proc. AAAI Conference on Artificial Intelligence*, 2021.

[13] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[14] Y. Freund and R. E. Schapire, "Adaptive game playing using multiplicative weights," *Games and Economic Behavior*, vol. 29, no. 1-2, pp. 79–103, 1999.

[15] E. Hazan, "Introduction to online convex optimization," *arXiv preprint arXiv:1909.05207*, 2019.

[16] K. C. Kalagarla, R. Jain, and P. Nuzzo, "Model-free reinforcement learning for optimal control of Markov decision processes under signal temporal logic specifications," *arXiv preprint arXiv:2109.13377*, 2021.

[17] J. Kivinen and M. K. Warmuth, "Exponentiated gradient versus gradient descent for linear predictors," *Information and Computation*, vol. 132, no. 1, pp. 1–63, 1997.