On the Uniqueness of Solution for the Bellman Equation of LTL Objectives

Zetong Xuan Z.XUAN@UFL.EDU
University of Florida, Gainesville, FL 32611, USA

Alper Kamil Bozkurt ALPER.BOZKURT@DUKE.EDU
Duke University, Durham, NC 27708, USA

Miroslav Pajic MIROSLAV.PAJIC@DUKE.EDU
Duke University, Durham, NC 27708, USA

Yu Wang YUWANG1@UFL.EDU
University of Florida, Gainesville, FL 32611, USA

Editors: A. Abate, K. Margellos, A. Papachristodoulou

Abstract

Surrogate rewards for linear temporal logic (LTL) objectives are commonly utilized in planning problems for LTL objectives. In a widely-adopted surrogate reward approach, two discount factors are used to ensure that the expected return approximates the satisfaction probability of the LTL objective. The expected return then can be estimated by methods using the Bellman updates such as reinforcement learning. However, the uniqueness of the solution to the Bellman equation with two discount factors has not been explicitly discussed. We demonstrate with an example that when one of the discount factors is set to one, as allowed in many previous works, the Bellman equation may have multiple solutions, leading to inaccurate evaluation of the expected return. We then propose a condition for the Bellman equation to have the expected return as the unique solution, requiring the solutions for states inside a rejecting bottom strongly connected component (BSCC) to be 0. We prove this condition is sufficient by showing that the solutions for the states with discounting can be separated from those for the states without discounting under this condition.

Keywords: Markov Chain, Limit-Deterministic Büchi Automaton, Reachability, Büchi Condition

1. Introduction

Modern autonomous systems need to solve planning problems for complex rule-based tasks that are usually expressible by linear temporal logic (LTL) (Pnueli, 1977). LTL is a symbolic language that helps fully automate the design process with computer algorithms. When the planning environment can be modeled by Markov decision processes (MDPs), the planning problems of finding the optimal policy to maximize the probability of achieving an LTL objective can be solved by model checking techniques (Baier and Katoen, 2008; Fainekos et al., 2005; Kress-Gazit et al., 2009).

However, the utility of model checking is limited when the transition probabilities of the MDP model are unknown. A promising solution, in such scenarios, is to deploy reinforcement learning (RL) (Sutton and Barto, 2018) to find the optimal policy from sampling. Early efforts in this direction have been confined to particular subsets of LTL (e.g. Li et al. (2017); Li and Belta (2019); Cohen and Belta (2023)), relied restricted semantics (e.g. Littman et al. (2017)), or assumed prior knowledge of the MDP's topology (e.g. Fu and Topcu (2014)) — understanding the presence or absence of transitions between any two given states. Model-based RL methods have also been applied by first estimating all the transitions of the MDP and applying model checking with a consideration on the estimation error (Brázdil et al., 2014). However, the computation complexity can be unnecessarily high since not all transitions are equally relevant (Ashok et al., 2019).

Recent works have used model-free RL for LTL objectives on MDPs with unknown transition probabilities (Sadigh et al., 2014; Hasanbeig et al., 2019; Hahn et al., 2020; Bozkurt et al., 2020). These approaches are all based on constructing ω-regular automata for the LTL objectives and translating the LTL objective into surrogate rewards within the product of the MDP and the automaton. The surrogate rewards yield the Bellman equations for the satisfaction probability of the LTL objective for a given policy, which can be solved through sampling by RL.

The first approach (Sadigh et al., 2014) employs Rabin automata to transform LTL objectives into Rabin objectives, which are then translated into surrogate rewards, assigning constant positive rewards to certain "good" states and negative rewards to "bad" states. However, this surrogate reward function is not technically correct, as demonstrated in (Hahn et al., 2019). The second approach (Hasanbeig et al., 2019) employs limit-deterministic Büchi automata to translate LTL objectives into surrogate rewards that assign a constant reward for "good" states with a constant discount factor. This approach is also technically flawed, as demonstrated by (Hahn et al., 2020). The third method (Bozkurt et al., 2020) also utilizes limit-deterministic Büchi automata but introduces surrogate rewards featuring a constant reward for "good" states and two discount factors that converge to 1 throughout the training process.

In more recent works (Voloshin et al., 2023; Shao and Kwiatkowska, 2023; Cai et al., 2021; Hasanbeig et al., 2023), the surrogate reward with two discount factors from (Bozkurt et al., 2020) was used while allowing one discount factor to be equal to 1. We noticed that in this case, the Bellman equation may have multiple solutions, as that discount factor of 1 does not provide contraction in many states for the Bellman operator. Consequently, the RL algorithm may not converge or may converge to a solution that deviates from the satisfaction probabilities of the LTL objective, leading to non-optimal policies. To illustrate this, we present a concrete example. To identify the satisfaction probabilities from the multiple solutions, we propose a sufficient condition that requires the solution of the Bellman equation to be 0 on all rejecting BSCCs, in which the discount factor is always 1.

We show that, under this sufficient condition, the Bellman equation has a unique solution that approximates the satisfaction probabilities for LTL objectives by the following procedure. When one of the discount factors equals 1, we partition the state space into states with discounting and states without discounting based on surrogate reward. In this case, we first characterize the relationship between all states with discounting and show that their solution is unique since the Bellman operator always has contractions in these states. Then, we show that the whole solution is unique since the solution on states without discounting is uniquely determined by states with discounting.

2. Preliminaries

This section introduces preliminaries on labeled Markov decision processes, linear temporal logic, and probabilistic model checking.

2.1. Labeled Markov Decision Processes

We use labeled Markov decision processes (LMDPs) to model planning problems where each decision has a potentially probabilistic outcome. LMDPs augment standard Markov decision processes

(Baier and Katoen, 2008) with state labels, enabling assigning properties, such as safety and liveness, to a sequence of states.

Definition 1 *A labeled Markov decision process is a tuple $\mathcal{M} = (S, A, P, s_{\text{init}}, \Lambda, L)$ where*

- *S is a finite set of states and $s_{\text{init}} \in S$ is the initial state,*
- *A is a finite set of actions where $A(s)$ denotes the set of allowed actions in the state $s \in S$,*
- *$P : S \times A \times S \to [0, 1]$ is the transition probability function such that for all $s \in S$, we have*

$$\sum_{s' \in S} P(s, a, s') = \begin{cases} 1, & a \in A(s) \\ 0, & a \notin A(s) \end{cases},$$

- *Λ is a finite set of atomic propositions and $L : S \to 2^{\Lambda}$ is a labeling function.*

A path of the LMDP $\mathcal{M}$ is an infinite state sequence $\sigma = s_0 s_1 s_2 \cdots$ such that for all $i \geq 0$, there exists $a_i \in A(s)$ and $s_i, s_{i+1} \in S$ with $P(s_i, a_i, s_{i+1}) > 0$. We can construct a corresponding semantic path as $L(\sigma) = L(s_0)L(s_1)\cdots$ by the labeling function $L(s)$. Given a path σ, the ith state is denoted by $\sigma[i] = s_i$. We denote the prefix by $\sigma[{:}i] = s_0 s_1 \cdots s_i$ and suffix by $\sigma[i{+}1{:}] = s_{i+1}s_{i+2}\cdots$.

2.2. Linear Temporal Logic and Limit-Deterministic Büchi Automata

In an LMDP $\mathcal{M}$, whether a given semantic path $L(\sigma)$ satisfies a property such as avoiding unsafe states can be expressed using Linear Temporal Logic (LTL). LTL can specify the change of labels along the path by connecting Boolean variables over the labels with two propositional operators, negation ($\neg$) and conjunction ($\wedge$), two temporal operators, next ($\bigcirc$) and until ($\cup$).

Definition 2 *The LTL formula is defined by the syntax*

$$\varphi ::= \text{true} \,|\, \alpha \,|\, \varphi_1 \wedge \varphi_2 \,|\neg\varphi \,|\, \bigcirc \varphi \,|\, \varphi_1 \cup \varphi_2, \alpha \in \Lambda \tag{1}$$

Satisfaction of an LTL formula φ on a path σ of an MDP (denoted by $\sigma \models \varphi$) is defined as, $\alpha \in \Lambda$ is satisfied on σ if $\alpha \in L(\sigma[1])$, $\bigcirc\varphi$ is satisfied on σ if φ is satisfied on $\sigma[1{:}]$, $\varphi_1 \cup \varphi_2$ is satisfied on σ if there exists i such that $\sigma[i{:}] \models \varphi_2$ and for all $j < i$, $\sigma[j{:}] \models \varphi_1$.

Other propositional and temporal operators can be derived from previous operators, e.g., (or) $\varphi_1 \vee \varphi_2 := \neg(\neg\varphi_1 \wedge \neg\varphi_2)$, (eventually) $\Diamond\varphi := \text{true} \cup \varphi$ and (always) $\Box\varphi := \neg\Diamond\neg\varphi$.

We can use Limit-Deterministic Büchi Automata (LDBA) to check the satisfaction of an LTL formula on a path.

Definition 3 *An LDBA is a tuple $\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, q_0, B)$ where $\mathcal{Q}$ is a finite set of automaton states, Σ is a finite alphabet, $\delta : \mathcal{Q} \times (\Sigma \cup \{\epsilon\}) \to 2^{\mathcal{Q}}$ is a (partial) transition function, q_0 is an initial state, and B is a set of accepting states, δ is total except for the ϵ-transitions ($|\delta(q, \alpha)| = 1$ for all $q \in \mathcal{Q}, \alpha \in \Sigma$), and there exists a bipartition of $\mathcal{Q}$ to an initial and an accepting component $\mathcal{Q}_{ini} \cup \mathcal{Q}_{acc} = \mathcal{Q}$ such that*

- *there is no transition from $\mathcal{Q}_{acc}$ to $\mathcal{Q}_{ini}$, i.e., for any $q \in \mathcal{Q}_{acc}, v \in \Sigma, \delta(q, v) \subseteq \mathcal{Q}_{acc}$,*

- *all the accepting states are in $\mathcal{Q}_{acc}$, i.e., $B \subseteq \mathcal{Q}_{acc}$,*
- *$\mathcal{Q}_{acc}$ does not have any outgoing ϵ-transitions, i.e., $\delta(q, \epsilon) = \emptyset$ for any $q \in \mathcal{Q}_{acc}$.*

A run is an infinite automaton transition sequence $\rho = (q_0, w_0, q_1), (q_1, w_1, q_2) \cdots$ such that for all $i \geq 0$, $q_{i+1} \in \delta(q_i, w_i)$. The run ρ is accepted by the LDBA if it satisfies the Büchi condition, i.e., $\inf(\rho) \cap B \neq \emptyset$, where $\inf(\rho)$ denotes the set of automaton states visited by ρ infinitely many times.

A path $\sigma = s_0 s_1 \dots$ of an LMDP $\mathcal{M}$ is considered accepted by an LDBA $\mathcal{A}$ if the semantic path $L(\sigma)$ is the corresponding word w of an accepting run ρ after elimination of ϵ-transitions.

Lemma 4 *(Sickert et al., 2016, Theorem 1) Given an LTL objective φ, we can construct an LDBA $\mathcal{A}_\varphi$ (with labels $\Sigma = 2^\Lambda$) such that a path $\sigma \models \varphi$ if and only if σ is accepted by the LDBA $\mathcal{A}_\varphi$.*

2.3. Product MDP

Planning problems for LTL objectives typically requires a (history-dependent) policy, which determines the current action based on all previous state visits.

Definition 5 *A policy π is a function $\pi : S^+ \to A$ such that $\pi(\sigma[:n]) \in A(\sigma[n])$, where S^+ stands for the set all non-empty finite sequences taken from S. A memoryless policy is a policy that only depends on the current state $\pi : S \to A$. Given a LMDP $\mathcal{M} = (S, A, P, s_0, \Lambda, L)$ and a memoryless policy π, a Markov chain (MC) induced by policy π is a tuple $\mathcal{M}_\pi = (S, P_\pi, s_0, \Lambda, L)$ where $P_\pi(s, s') = P(s, \pi(s), s')$ for all $s, s' \in S$.*

Using the LDBA, we construct a product MDP that augments the MDP state space along with the state space of the LDBA, such that the state of the product MDP encodes both the physical state and the progression of the LTL objective. In this manner, we "lift" the planning problem to the product MDP. Given that the state of the product MDP now encodes all the information necessary for planning, the action can be determined by the current state of the product MDP, resulting in history-independent policies Formally, the product MDP is defined as follows:

Definition 6 *A product MDP $\mathcal{M}^\times = (S^\times, A^\times, P^\times, s_0^\times, B^\times)$ of an LMDP $\mathcal{M} = (S, A, P, s_0, \Lambda, L)$ and an LDBA $\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, q_0, B)$ is defined by the set of states $S^\times = S \times \mathcal{Q}$, the set of actions $A^\times = A \cup \{\epsilon_q | q \in \mathcal{Q}\}$, the transition probability function*

$$P^\times(\langle s, q\rangle, a, \langle s', q'\rangle) = \begin{cases} P(s, a, s') & q' = \delta(q, L(s)), a \notin A^\epsilon \\ 1 & a = \epsilon_{q'}, q' \in \delta(q, \epsilon), s = s' \\ 0 & \text{otherwise} \end{cases},$$

the initial state $s_0^\times = \langle s_0, q_0\rangle$, and the set of accepting states $B^\times = \{\langle s, q\rangle \in S^\times | q \in B\}$. We say a path σ satisfies the Büchi condition φ_B if $\inf(\sigma) \cap B^\times \neq \emptyset$. Here, $\inf(\sigma)$ denotes the set of states visited infinitely many times on σ.

The transitions of the product MDP $\mathcal{M}^\times$ are derived by combining the transitions of the MDP $\mathcal{M}$ and the LDBA $\mathcal{A}$. Specifically, the multiple ϵ-transitions starting from the same states in the LDBA are differentiated by their respective end states q and are denoted as ϵ_q. These ϵ-transitions in the LDBA give rise to corresponding ϵ-actions in the product MDP, each occurring with a probability of 1. The limit-deterministic nature of LDBAs ensures that the presence of these ϵ-actions within the product MDPs does not prevent the quantitative analysis of the MDPs for planning. In other words, any optimal policy for a product MDP induces an optimal policy for the original MDP, as formally stated below.

Lemma 7 (Sickert et al. (2016)) *For given an LMDP $\mathcal{M}$ and an LTL objective φ, let $\mathcal{A}_\varphi$ be the LDBA derived from φ and let $\mathcal{M}^\times$ be the product MDP constructed from $\mathcal{M}$ and $\mathcal{A}_\varphi$, with the set of accepting states $B^\times$. Then, a memoryless policy $\pi^\times$ that maximizes the probability of satisfying the Büchi condition on $\mathcal{M}^\times$, $P_{\sigma^\times}\big(\sigma^\times \models \square\lozenge B^\times\big)$ where $\sigma^\times \sim \mathcal{M}^\times_{\pi^\times}$, induces a finite memory policy π that maximizes the satisfaction probability $P_{\sigma\sim\mathcal{M}_\pi}\big(\sigma \models \varphi\big)$ on $\mathcal{M}$.*

3. Problem Formulation

In the previous section, we have shown LTL objectives on an LMDP can be converted into a Büchi condition on the Product MDP. In this section, we focus on a common surrogate reward used for Büchi condition proposed in (Bozkurt et al., 2020) and study the uniqueness of solution for the Bellman equation of this surrogate reward, which has not been sufficiently discussed in previous work (Voloshin et al., 2023; Hasanbeig et al., 2023; Shao and Kwiatkowska, 2023).

For simplicity, we drop × from the product MDP notation and define the satisfaction probability for the Büchi condition as

$$P(s \models \square\lozenge B) := P_{\sigma\sim\mathcal{M}_\pi}\big(\sigma \models \square\lozenge B \mid \exists t : \sigma[t] = s\big). \tag{2}$$

When the product MDP model is unknown, the traditional model-based method through graph search (Baier and Katoen, 2008) is not applicable. Alternatively, we may use model-free reinforcement learning with a two-discount-factor surrogate reward proposed by (Bozkurt et al., 2020) and widely used in (Voloshin et al., 2023; Shao and Kwiatkowska, 2023; Cai et al., 2021; Hasanbeig et al., 2023; Cai et al., 2023). It consists of a reward function $R : S \to \mathbb{R}$ and a state-dependent discount factor function $\Gamma : S \to (0, 1]$ with $0 < \gamma_B < \gamma \le 1$,

$$R(s) := \begin{cases} 1-\gamma_B & s \in B \\ 0 & s \notin B \end{cases}, \quad \Gamma(s) := \begin{cases} \gamma_B & s \in B \\ \gamma & s \notin B \end{cases}. \tag{3}$$

A positive reward is collected only when an accepting state is visited along the path. Suppose the discount factor $\gamma = 1$; then, the satisfaction of Büchi condition results in a summation of a geometric series equal to one. The probability of whether a path satisfies the Büchi condition is equal to how likely such a geometric series exists along a path.

For this surrogate reward, the K-step return ($K \in \mathbb{N}$ or $K = \infty$) of a path from time $t \in \mathbb{N}$ is

$$G_{t:K}(\sigma) = \sum_{i=0}^{K} R(\sigma[t+i]) \cdot \prod_{j=0}^{i-1} \Gamma(\sigma[t+j]), \quad G_t(\sigma) = \lim_{K\to\infty} G_{t:K}(\sigma). \tag{4}$$

Accordingly, the value function $V_\pi(s)$ is the expected return conditional on the path starting at s under the policy π. It approximates the satisfaction probability thus serves as a metric for policy.

$$\begin{aligned} V_\pi(s) &= \mathbb{E}_\pi[G_t(\sigma) \,|\, \sigma[t] = s] \\ &= \mathbb{E}_\pi[G_t(\sigma) \mid \sigma[t] = s, \sigma \models \square\lozenge B] \cdot P(s \models \square\lozenge B) \\ &+ \mathbb{E}_\pi[G_t(\sigma) \mid \sigma[t] = s, \sigma \not\models \square\lozenge B] \cdot P(s \not\models \square\lozenge B) \end{aligned} \tag{5}$$

Given a policy, the value function satisfies the Bellman equation.[1] The Bellman equation is derived from the fact that the value of the current state is equal to the expectation of the current reward plus the discounted value of the next state. For the surrogate reward in the equation (3), the Bellman equation is given as follows:

$$V_\pi(s) = \begin{cases} 1 - \gamma_B + \gamma_B \sum_{s' \in S} P_\pi(s, s') V_\pi(s') & s \in B \\ \gamma \sum_{s' \in S} P_\pi(s, s') V_\pi(s) & s \notin B \end{cases}. \tag{6}$$

Previous work (Voloshin et al., 2023; Hasanbeig et al., 2023; Shao and Kwiatkowska, 2023) allows $\gamma = 1$. However, setting $\gamma = 1$ can cause multiple solutions to the Bellman equations, raising concerns about applying model-free RL. This motivates us to study the following problem.

> **Problem Formulation:** For given (product) MDP $\mathcal{M}$ from Definition 6 and the surrogate reward from (3), and a policy π, find the sufficient conditions under which the Bellman equation from (6) has a unique solution.

The following example shows the Bellman equation (6) has multiple solutions when $\gamma = 1$ (3). An incorrect solution, different than the expected return from (5), hinders accurate policy evaluation and restricts the application of RL and other optimization techniques.

Example 1 *Consider a (product) MDP with three states $S = \{s_1, s_2, s_3\}$ where s_1 is the initial state and $B = \{s_2\}$ is the set of accepting states as shown in Figure 1. In s_1, the action α leads to s_2 and the action β leads to s_3. Since s_2 is the only accepting state, α is the optimal action that maximizing the expected return. However, there exists a solution to the corresponding Bellman equation suggesting β is the optimal action, as follows:*

$$a^* := \underset{a \in \{\alpha, \beta\}}{\operatorname{argmax}} \{P(s, a, s') V(s')\} = \underset{a \in \{\alpha, \beta\}}{\operatorname{argmax}} \begin{cases} V(s_2) & \textit{if } a = \alpha, \\ V(s_3) & \textit{if } a = \beta, \end{cases} \tag{7}$$

where $V(s_2)$ and $V(s_3)$ can be computed using the Bellman equation (3) as the following:

$$V(s_2) = 1 - \gamma_B + \gamma_B V(s_2), \quad V(s_3) = V(s_3). \tag{8}$$

yielding $V(s_2) = 1$ and $V(s_3) = c$ where $c \in \mathbb{R}$ is an arbitrary constant. Suppose $c = 2$ is chosen as the solution, then the optimal action will be incorrectly identified as β by (7).

Remark 8 *The product MDP from Definition 6 is exactly an MDP in the general sense. The surrogate reward (3) and our result based on it work for the general MDPs with Büchi objectives.*

4. Overview of Main Results

The Bellman equation provides a necessary condition for determining the value function. However, it can have several solutions, with only one being the actual value function (for instance, the Bellman equation for reachability (Baier and Katoen, 2008, P851)). It is crucial to identify conditions that eliminate incorrect solutions since solving techniques like model-free RL may struggle to converge or converge to an incorrect solution in the presence of multiple solutions.

[1] We call $V_\pi(s) = R(s) + \gamma \sum_{s' \in S} P_\pi(s, s') V_\pi(s')$ as the Bellman equation and $V_\pi^*(s) = \max_{a \in A(s)} \{R(s) + \gamma \sum_{s' \in S} P(s, a, s') V_\pi^*(s')\}$ as the Bellman optimality equation.

In Example 1, for $c = 0$, the solution for $V(s_3)$ is the value function equal to zero since no reward will be collected on this self-loop based on (3). Generally, the solution should be zero for all states in the rejecting BSCCs, as defined below.

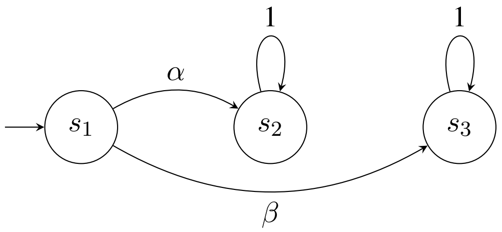

Figure 1: Example of a three-state Markov decision process. The resulting Bellman equation (8) has multiple solutions when $\gamma = 1$ in the surrogate reward (3), which can mislead to the suboptimal actions.

Definition 9 *A bottom strongly connected component (BSCC) of an MC is a strongly connected component without outgoing transitions. A BSCC is rejecting*[2] *if all states $s \notin B$. Otherwise, we call it an accepting BSCC.*

By Definition 9, there will not be any accepting states visited on a path starting from a state in the rejecting BSCCs. Thus, the value function for all states in the rejecting BSCCs equals 0 based on (3). Setting the values for all states within a rejecting BSCC to zero is a sufficient condition for the Bellman equation solution equaling the value function, as stated below.

Theorem 10 *The Bellman equation (6) has the value function as the unique solution, if and only if i) the discount factor $\gamma < 1$ or ii) the discount factor $\gamma = 1$ and the solution for any state in a rejecting BSCC is zero.*

5. Methodology

We illustrate the proof of Theorem 10 in this section and provide detailed proofs in the extended version (Xuan et al., 2023) . Specifically, we first prove it for the case of $\gamma < 1$ and then move to the case of $\gamma = 1$. The surrogate reward (3) depends on whether a state is an accepting state or not. Thus, we split the state space S by the accepting states B and rejecting states $\neg B := S \backslash B$. The Bellman equation can be rewritten in the following form,

$$\begin{bmatrix} V^B \\ V^{\neg B} \end{bmatrix} = (1-\gamma_B)\begin{bmatrix} \mathbb{I}_m \\ \mathbb{O}_n \end{bmatrix} + \underbrace{\begin{bmatrix} \gamma_B I_{m\times m} & \\ & \gamma I_{n\times n} \end{bmatrix}}_{\Gamma_B} \underbrace{\begin{bmatrix} P_{\pi,B\to B} & P_{\pi,B\to\neg B} \\ P_{\pi,\neg B\to B} & P_{\pi,\neg B\to\neg B} \end{bmatrix}}_{P_\pi} \begin{bmatrix} V^B \\ V^{\neg B} \end{bmatrix}, \tag{9}$$

where $m = |B|$, $n = |\neg B|$, $V^B \in \mathbb{R}^m$, $V^{\neg B} \in \mathbb{R}^n$ are the vectors listing the value function for all $s \in B$ and $s \in \neg B$, respectively. $\mathbb{I}$ and $\mathbb{O}$ are column vectors with all 1 and 0 elements, respectively. Each of the matrices $P_{\pi,B\to B}$, $P_{\pi,B\to\neg B}$, $P_{\pi,\neg B\to B}$, $P_{\pi,\neg B\to\neg B}$ contains the transition probability from a set of states to a set of states, their combination is the transition matrix P_π for the induced MC. In the following, we assume a fixed policy π, leading us to omit the π subscript from most notation when its implication is clear from the context.

[2] Here we call a state $s \in B$ as an accepting state, a state $s \notin B$ as a rejecting state. Notice that an accepting state must not exist in a rejecting BSCC and a rejecting state may exist in an accepting BSCC.

5.1. The case $\gamma < 1$

Proposition 11 *If $\gamma < 1$ in the surrogate reward (3), then the Bellman equation (9) has the value function as the unique solution.*

As $\gamma < 1$, the invertibility of $(I - \Gamma_B P_\pi)$ can be shown by applying Gershgorin circle theorem (Bell, 1965, Theorem 0). Any eigenvalue λ of $\Gamma_B P_\pi$ satisfies $|\lambda| < 1$ since each row sum of $\Gamma_B P_\pi$ is strictly less than 1. Then, the solution for the Bellman equation (9) can be uniquely determined as

$$\begin{bmatrix} V^B \\ V^{\neg B} \end{bmatrix} = (1 - \gamma_B)(I_{m+n} - \Gamma_B P_\pi)^{-1} \begin{bmatrix} \mathbb{I}_m \\ \mathbb{O}_n \end{bmatrix}. \tag{10}$$

5.2. The case $\gamma = 1$

For $\gamma = 1$, the matrix $(I - \Gamma_B P_\pi)$ may not be invertible, causing the Bellman equation (9) to have multiple solutions. Since the solution may not be the value function here, we use $U^B \in \mathbb{R}^m$ and $U^{\neg B} \in \mathbb{R}^n$ to represent a solution on states in B and $\neg B$, respectively. In an induced MC, a path starts in an initial state, travels finite steps among the transient states, and eventually enters a BSCC. If the induced MC has only accepting BSCCs, the connection between all states in B can be captured by a new transition matrix, and the Bellman operator is contractive on the states in B. Thus, we can show the solution is unique in all the states in Section 5.2.1. In the general case where rejecting BSCCs also exists in the MC, we introduce a sufficient condition of fixing all solutions within rejecting BSCCs to zero. We demonstrate the uniqueness of the solution under this condition first on U^B and then on $U^{\neg B}$ in Section 5.2.2.

5.2.1. When the MC only has accepting BSCCs

This section focuses on proving that the Bellman equation (9) has a unique solution when there are no rejecting BSCCs in the MC. The result is as follows,

Proposition 12 *If the MC only has accepting BSCCs and $\gamma = 1$ in the surrogate reward (3), then the Bellman equation (9) has a unique solution $[{U^B}^T, {U^{\neg B}}^T]^T = \mathbb{I}$.*

The intuition behind the proof is to capture the connection between all states B by a new transition matrix P_π^B, and using $I - \gamma_B P_\pi^B$ invertible to show the solutions U_B is unique. Then, we show $U^{\neg B}$ is uniquely determined by U^B.

We start with constructing a transition matrix P_π^B for the states in B whose (i, j)th element, denoted by $(P_\pi^B)_{ij}$, is the probability of visiting jth state in B without visiting any state in B after leaving the ith state in B.

$$P_\pi^B := P_{\pi,B\to B} + P_{\pi,B\to\neg B} \sum_{k=0}^{\infty} P^k_{\pi,\neg B\to\neg B} P_{\pi,\neg B\to B}. \tag{11}$$

In (11), the matrix element $(P^k_{\pi,\neg B\to\neg B})_{ij}$ represents the probability of a path leaving the state i and visiting state j after k steps without travelling through any states in B. However, the absence of rejecting BSCCs ensures that any path will visit a state in B in finite times with probability 1. Thus, for any $i, j \in \neg B$, $\lim_{k\to\infty}(P^k_{\pi,\neg B\to\neg B})_{ij} = 0$. This limit implies any eigenvalue λ of $P_{\pi,\neg B\to\neg B}$ satisfies $|\lambda| < 1$ and therefore $\sum_{k=0}^{\infty} P^k_{\pi,\neg B\to\neg B}$ can be replaced by $(I - P_{\pi,\neg B\to\neg B})^{-1}$ in (11),

$$P^B_\pi = P_{\pi,B\to B} + P_{\pi,B\to\neg B}(I - P_{\pi,\neg B\to\neg B})^{-1} P_{\pi,\neg B\to B}. \tag{12}$$

Since all the elements on the right-hand side are greater or equal to zero, for any $i, j \in B$, $(P^B_\pi)_{ij} \geq 0$. Since there are only accepting BSCCs in the MC, given a path starting from an arbitrary state in B, the path will visit an accepting state in finite steps with probability one, ensuring that for all $i \in B$, $\sum_{j\in S}(P^B_\pi)_{ij} = 1$. Thus, P^B_π is a probability matrix that can be used to describe the behaviour of an MC with the state space as B only.

Remark 13 *For a given MC with only accepting BSCCs $\mathcal{M}_\pi = (S, P_\pi, s_0, \Lambda, L)$, we can construct an MC consisting of the accepting states $\mathcal{M}^B_\pi := (B, P^B_\pi, \mu, \Lambda, L)$. This new MC, referred to as the accepting MC, has the state space defined as the set of accepting states B. The transition probabilities P^B_π (from (12)) are the transition probabilities between the accepting states in $\mathcal{M}_\pi$. The initial distribution μ is a distribution on B and determined by s_0 as follows:*

$$\text{if } s_0 \in B, \quad \lambda(s) = \begin{cases} 1 & s = s_0 \\ 0 & s \neq s_0 \end{cases}, \quad \text{if } s_0 \notin B, \quad \mu(s) = (P_{init})_{s_0\, s} \tag{13}$$

where $P_{init} := (I - P_{\pi,\neg B\to\neg B})^{-1} P_{\pi,\neg B\to B}$ is a matrix. Each element $(P_{init})_{ij}$ represents the probability of a path leaving the state $i \in \neg B$ and visiting state $j \in B$ without visiting any state in B between the leave and visit. Since the absence of rejecting BSCC, we can construct an accepting MC, and every state will have a reward of $1 - \gamma_B$ and a discount factor of γ_B.

Lemma 14 *Suppose there is no rejecting BSCC, for $\gamma = 1$ in (3), the Bellman equation (9) is equivalent to the following form,*

$$\begin{bmatrix} U^B \\ U^{\neg B} \end{bmatrix} = (1 - \gamma_B) \begin{bmatrix} \mathbb{I}_m \\ \mathbb{O}_n \end{bmatrix} + \begin{bmatrix} \gamma_B I_{m\times m} & \\ & I_{n\times n} \end{bmatrix} \begin{bmatrix} P^B_\pi & \\ P_{\pi,\neg B\to B} & P_{\pi,\neg B\to\neg B} \end{bmatrix} \begin{bmatrix} U^B \\ U^{\neg B} \end{bmatrix}. \tag{14}$$

The equation (14) implies that the solution U^B does not rely on the rejecting states $\neg B$. Subsequently, we leverage the fact that $U^{\neg B}$ is uniquely determined by U^B to establish the uniqueness of the overall solution V.

Proposition 12 shows the solutions for the states inside an accepting BSCC have to be 1. All states outside the BSCC cannot be reached from a state inside the BSCC, thus the solution for states outside this BSCC is not involved in the solution for states inside the BSCC. By Lemma 14, the Bellman equation for an accepting BSCC can be rewritten into the form of (14) where U_B and $U_{\neg B}$ stands for the solution for accepting states and rejecting states inside this BSCC, and vector $\mathbb{I}$ is the unique solution.

5.2.2. When accepting and rejecting BSCC both exist in the MC

Having established the uniqueness of solutions in the case of accepting BSCCs, we now shift our focus to the general case involving rejecting BSCCs. We state in Proposition 12 that the solutions for the states in the accepting BSCCs are unique and equal to $\mathbb{I}$. We now demonstrate that setting the solutions for the states in rejecting BSCCs to $\mathbb{O}$ ensures the uniqueness and correctness of the solutions for all states. We partition the state space further into $\{B_A, B_T, \neg B_A, \neg B_R, \neg B_T\}$, where B_A is the set of accepting states in the BSCCs, $B_T := B \backslash B_A$ is the set of transient accepting states. $\neg B_A$ is the set of rejecting states in the accepting BSCCs, $\neg B_R$ is the set of rejecting states in the rejecting BSCCs, and $\neg B_T := \neg B \backslash (\neg B_A \cup \neg B_R)$ is set of transient rejecting states. We rewrite the Bellman equation (9) in the following form,

$$\begin{bmatrix} U^{B_T} \\ U^{B_A} \\ U^{\neg B_T} \\ U^{\neg B_A} \\ U^{\neg B_R} \end{bmatrix} = (1-\gamma_B)\begin{bmatrix} \mathbb{I}_m \\ \mathbb{O}_n \end{bmatrix} + \begin{bmatrix} \gamma_B I_{m\times m} & \\ & I_{n\times n} \end{bmatrix}$$

$$\begin{bmatrix} P_{\pi,B_T\to B_T} & P_{\pi,B_T\to B_A} & P_{\pi,B_T\to\neg B_T} & P_{\pi,B_T\to\neg B_A} & P_{\pi,B_T\to\neg B_R} \\ & P_{\pi,B_A\to B_A} & & P_{\pi,B_A\to\neg B_A} & P_{\pi,B_A\to\neg B_R} \\ P_{\pi,\neg B_T\to B_T} & P_{\pi,\neg B_T\to B_A} & P_{\pi,\neg B_T\to\neg B_T} & P_{\pi,\neg B_T\to\neg B_A} & P_{\pi,\neg B_T\to\neg B_R} \\ & P_{\pi,\neg B_A\to B_A} & & P_{\pi,\neg B_A\to\neg B_A} & P_{\pi,\neg B_A\to\neg B_R} \\ & & & & P_{\pi,\neg B_R\to\neg B_R} \end{bmatrix}\begin{bmatrix} U^{B_T} \\ U^{B_A} \\ U^{\neg B_T} \\ U^{\neg B_A} \\ U^{\neg B_R} \end{bmatrix}. \tag{15}$$

The solution for states inside BSCCs has been fixed as $[{U^{B_A}}^T, {U^{\neg B_A}}^T]^T = \mathbb{I}$ and $U^{\neg B_R} = \mathbb{O}$. The solution U^{B_T} and $U^{\neg B_T}$ for transient states remain to be shown as unique. We rewrite the Bellman equation (15) into the following form (16) where U^{B_T} and $U^{\neg B_T}$ are the only variables,

$$\begin{bmatrix} U^{B_T} \\ U^{\neg B_T} \end{bmatrix} = \begin{bmatrix} \gamma_B I_{m_1\times m_1} & \\ & I_{n_1\times n_1} \end{bmatrix}\begin{bmatrix} P_{\pi,B_T\to B_T} & P_{\pi,B_T\to\neg B_T} \\ P_{\pi,\neg B_T\to B_T} & P_{\pi,\neg B_T\to\neg B_T} \end{bmatrix}\begin{bmatrix} U^{B_T} \\ U^{\neg B_T} \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \tag{16}$$

here $m_1 = |U^{B_T}|$, $n_1 = |U^{\neg B_T}|$ and

$$\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = (1-\gamma_B)\begin{bmatrix} \mathbb{I}_{m_1} \\ \mathbb{O}_{n_1} \end{bmatrix} + \begin{bmatrix} \gamma_B I_{m_1\times m_1} & \\ & I_{n_1\times n_1} \end{bmatrix}\begin{bmatrix} P_{\pi,B_T\to B_A} & P_{\pi,B_T\to\neg B_A} \\ P_{\pi,\neg B_T\to B_A} & P_{\pi,\neg B_T\to\neg B_A} \end{bmatrix}\begin{bmatrix} \mathbb{I}_{m_1} \\ \mathbb{I}_{n_1} \end{bmatrix}.$$

Lemma 15 *The equation* (16) *has a unique solution.*

We demonstrate U^{B_T} does not rely on states in $\neg B_T$ and $U^{\neg B_T}$ is uniquely determined by U^{B_T}. Then the uniqueness of U^{B_T} can be shown first, consequently, uniqueness of $U^{\neg B_T}$ can be shown.

Proof for Theorem 10 For the case $\gamma = 1$, we have shown that the equation (16) with surrogate reward (3) has a unique solution in Lemma 15. In order to complete the proof for theorem 10, what remains to be shown is the unique solution of the equation (16) is equal to the value function (5).

The solution to the equation (16) is unique. For all $s \in \neg B_R$, the value function $V(s) = 0$, then the value function is the unique solution for equation (16). Under the condition that the solution for all rejecting BSCCs is zero, the Bellman equation (9) is equivalent to the equation (16). We can say theorem 10 is true. ■

6. Conclusion

This work uncovers a challenge when using surrogate rewards with two discount factors for LTL objectives, which has been unfortunately overlooked by many previous works. Specifically, we show setting one of the discount factors to one can cause the Bellman equation to have multiple solutions, hindering the derivation of the value function. We discuss the uniqueness of the solution for the Bellman Equation with two discount factors and propose a condition to identify the value function from the multiple solutions.

References

Pranav Ashok, Jan Křetínský, and Maximilian Weininger. PAC Statistical Model Checking for Markov Decision Processes and Stochastic Games. In *Computer Aided Verification*, pages 497–519. Springer International Publishing, 2019.

Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2008.

Howard E. Bell. Gershgorin's Theorem and the Zeros of Polynomials. *The American Mathematical Monthly*, 72(3):292–295, 1965.

Alper Kamil Bozkurt, Yu Wang, Michael M. Zavlanos, and Miroslav Pajic. Control Synthesis from Linear Temporal Logic Specifications using Model-Free Reinforcement Learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10349–10355, 2020.

Tomáš Brázdil, Krishnendu Chatterjee, Martin Chmelík, Vojtěch Forejt, Jan Křetínský, Marta Kwiatkowska, David Parker, and Mateusz Ujma. Verification of Markov Decision Processes Using Learning Algorithms. In *Automated Technology for Verification and Analysis*, pages 98–114. Springer International Publishing, 2014.

Mingyu Cai, Mohammadhosein Hasanbeig, Shaoping Xiao, Alessandro Abate, and Zhen Kan. Modular deep reinforcement learning for continuous motion planning with temporal logic. *IEEE Robotics and Automation Letters*, 6(4):7973–7980, 2021.

Mingyu Cai, Shaoping Xiao, Junchao Li, and Zhen Kan. Safe reinforcement learning under temporal logic with reward design and quantum action selection. *Scientific Reports*, 13(1):1925, 2023.

Max Cohen and Calin Belta. Temporal logic guided safe model-based reinforcement learning. In *Adaptive and Learning-Based Control of Safety-Critical Systems*, pages 165–192. Springer International Publishing, 2023.

G.E. Fainekos, H. Kress-Gazit, and G.J. Pappas. Temporal Logic Motion Planning for Mobile Robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2020–2025. IEEE, 2005.

Jie Fu and Ufuk Topcu. Probably approximately correct MDP learning and control with temporal logic constraints. 2014. https://doi.org/10.48550/arXiv.1404.7073.

Ernst Moritz Hahn, Mateo Perez, Sven Schewe, Fabio Somenzi, Ashutosh Trivedi, and Dominik Wojtczak. Omega-regular objectives in model-free reinforcement learning. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 395–412. Springer International Publishing, 2019.

Ernst Moritz Hahn, Mateo Perez, Sven Schewe, Fabio Somenzi, Ashutosh Trivedi, and Dominik Wojtczak. Faithful and effective reward schemes for model-free reinforcement learning of omega-regular objectives. In *Automated Technology for Verification and Analysis: 18th International Symposium, ATVA 2020, Hanoi, Vietnam, October 19–23, 2020, Proceedings*, pages 108–124. Springer-Verlag, 2020.

Hosein Hasanbeig, Daniel Kroening, and Alessandro Abate. Certified reinforcement learning with logic guidance. *Artificial Intelligence*, 322(C), 2023.

M. Hasanbeig, Y. Kantaros, A. Abate, D. Kroening, G. J. Pappas, and I. Lee. Reinforcement learning for temporal logic control synthesis with probabilistic satisfaction guarantees. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 5338–5343, 2019.

H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas. Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics*, 25(6):1370–1381, 2009.

Xiao Li and Calin Belta. Temporal logic guided safe reinforcement learning using control barrier functions, 2019. https://doi.org/10.48550/arXiv.1903.09885.

Xiao Li, Cristian-Ioan Vasile, and Calin Belta. Reinforcement learning with temporal logic rewards. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3834–3839, 2017.

Michael L. Littman, Ufuk Topcu, Jie Fu, Charles Isbell, Min Wen, and James MacGlashan. Environment-independent task specifications via GLTL, 2017. https://doi.org/10.48550/arXiv.1704.04341.

Amir Pnueli. The temporal logic of programs. In *Annual Symposium on Foundations of Computer Science*, 1977.

Dorsa Sadigh, Eric S. Kim, Samuel Coogan, S. Shankar Sastry, and Sanjit A. Seshia. A learning based approach to control synthesis of Markov decision processes for linear temporal logic specifications. In *53rd IEEE Conference on Decision and Control*, pages 1091–1096, 2014.

Daqian Shao and Marta Kwiatkowska. Sample Efficient Model-free Reinforcement Learning from LTL Specifications with Optimality Guarantees. In *Thirty-Second International Joint Conference on Artificial Intelligence*, volume 4, pages 4180–4189, 2023.

Salomon Sickert, Javier Esparza, Stefan Jaax, and Jan Křetínský. Limit-Deterministic Büchi Automata for Linear Temporal Logic. In *Computer Aided Verification*, volume 9780, pages 312–332. Springer International Publishing, 2016.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition edition, 2018.

Cameron Voloshin, Abhinav Verma, and Yisong Yue. Eventual Discounting Temporal Logic Counterfactual Experience Replay. In *Proceedings of the 40th International Conference on Machine Learning*, pages 35137–35150. PMLR, 2023.

Zetong Xuan, Alper Kamil Bozkurt, Miroslav Pajic, and Yu Wang. On the Uniqueness of Solution for the Bellman Equation of LTL Objectives. 2023. URL https://faculty.eng.ufl.edu/smart-autonomy-lab/wp-content/uploads/sites/392/2023/12/Unique_Solution_for_the_Bellman_Equation_of_LTL_Objectives.pdf.