2020 59th IEEE Conference on Decision and Control (CDC)
Jeju Island, Republic of Korea, December 14-18, 2020

Censored Markov Decision Processes: A Framework for Safe Reinforcement Learning in Collaboration with External Systems

Masahiro Kohjima, Masami Takahashi, and Hiroyuki Toda
NTT Service Evolution Laboratories, NTT Corporation, Yokosuka, Japan
{masahiro.kohjima.ev,masami.takahashi.xh,hiroyuki.toda.xb}@hco.ntt.co.jp

***Abstract*— The importance of safe reinforcement learning (safe RL) is widely recognized for enhancing real world systems. In this study, we construct the censored Markov decision process (CeMDP), a new Markov Decision Process (MDP) framework that describes the interaction of environment, learner and external systems, e.g., human intervention or pre-designed controller for emergency response. We also theoretically analyze the relation of CeMDP to existing frameworks such as the semi-Markov decision process, MDP with Option (OMDP) and standard MDP; the analysis clarifies that CeMDP is a special case of OMDP and can, with environment redefinition, be represented by MDP. This finding allows us to design planning and reinforcement learning algorithms for CeMDP. We confirm the validity of the theory and algorithms by numerical experiments.**

I. INTRODUCTION

The Markov decision process (MDP) is a versatile framework for sequential decision making [1] and is widely used in reinforcement learning (RL) [2]. Triggered by big success of RL in game AI [3], its application to real world systems is spreading to fields such as ridesharing dispatch [4], cellular network operation [5], traffic signal control [6], [7], and chemical plant control [8]. Given the social importance of these systems, *safe RL*, which tries to learn the best policy without behavior that may seriously damage the system, is gathering attention. Many approaches have been investigated up to now, e.g., methods which adopt new optimization criteria such as worst case criterion [9] and risk sensitive criterion [9], [10], [11], [12], and methods for safe-exploration using teacher's advice [13], [14] or control theory such as Lyapunov stability [15] and reachability [16]. See also [17] and references therein.

The motivation of this study is to consider an MDP framework using an external system for avoiding catastrophic behavior. The external system corresponds to e.g., human intervention and pre-designed action sequences for avoiding serious damage, so this may be classified as adopting teacher's advice, similar to [13], [14]. We consider that this framework is suited to safe RL and also that the RL learner must collaborate with humans to achieve the system's goal. Let us consider the RL usage examples of chemical plant control and autonomous vehicle control. In order to avoid failures due to overloading one or more devices in the plant or traffic accidents due to malfunction of on-vehicle cameras, it seems to be effective to turn control over to the human or a pre-designed response plan when the plant or vehicle experience an abnormal condition. We believe this type of man-machine collaboration is an important topic in control and AI communities.

In this study, we construct a new MDP framework, which we call *censored MDP* (CeMDP); it describes the interaction of environment, learner and external systems, e.g., human intervention and pre-designed action sequences. Figure 1 shows this interaction. Depending on the current state, the decision maker will be either the learner or the external system. The reason we call it *censored MDP* is because (i) control by the learner is censored and is allowed only when the system is in one of a known subset of states, and (ii) CeMDP has a strong relation to the censored Markov chain (CMC) [18], [19], [20], as is explained later.

We also theoretically analyze the relation of CeMDP to existing frameworks such as the semi-Markov decision process (semi-MDP) [21], MDP with option (OMDP) [22], and standard MDP. Our analysis clarifies that CeMDP is a special case of MDP with option (macro action) and CeMDP can also be taken as a variant of the standard MDP whose state space, transition probability and reward function are redefined. This finding allows us to design various planning and reinforcement learning algorithms for CeMDP. Here we show the CeMDP version of Q-Learning and that of value iteration.

The work most related to ours is the study of Saunders, et al. [14] which considers the MDP wherein agent action is modified by human intervention with an additional penalty reward being given to the agent. Although its effectiveness has been empirically confirmed, their work did not provide any theoretical relationship between their modified MDP and the original MDP and the other MDPs. In contrast, CeMDP, which we construct and analyze in this paper, differs from the MDP of Saunders et al. and a full theoretical relationship to existing frameworks is given.

The contributions of this paper are summarized below:

- We define CeMDP, a framework for safe RL that describes the interaction between agent, external system and environment.
- We clarify the relationship of CeMDP to existing MDP frameworks; CeMDP is a special case of OMDP and can be converted into MDP by redefining the environment.
- We introduce a planning and reinforcement learning algorithm that can be used for CeMDP and confirm its effectiveness by the numerical experiments.

考虑外部系统、很多理论证明

safe RL的一些方法：最差情况危险敏感安全探索Lyapunov控制，可达

作者把自己提出的CeMDP和传统的MDP关联起来

和作者最相关的研究：通过人类的介入给agent加入了一个惩罚的奖励，但是和本文相比没有和传统的MDP联系起来

978-1-7281-7447-1/20/$31.00 ©2020 IEEE

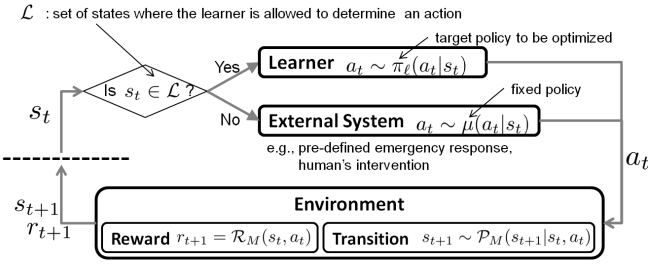


Fig. 1: Interaction of learner, external system and environment in Censored MDP (CeMDP).

II. Preliminaries

A. Markov Decision Processes

MDP is defined by the quintuplet $\{\mathcal{S}, \mathcal{A}, \mathcal{P}_M, \mathcal{R}_M, \gamma\}$, where $\mathcal{S} = \{1, 2, \cdots, |\mathcal{S}|\}$ is a finite set of *states* and $\mathcal{A} = \{1, 2, \cdots, |\mathcal{A}|\}$ is a set of *actions*. $\mathcal{P}_M : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ indicates *transition probabilities* and we denote the probability of moving from state s to state s' when action a is executed as $\mathcal{P}_M(s'|s, a)$. $\mathcal{R}_M : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a *reward function* and we denote the reward received when action a is executed in state s as $\mathcal{R}_M(s, a)$. $\gamma \in [0, 1)$ is a *discount factor*. We also use the matrix/vector representation of $\mathcal{P}_M$ and $\mathcal{R}_M$ given action a as $\boldsymbol{P}^a \in [0, 1]^{|\mathcal{S}|\times|\mathcal{S}|}$ and $\boldsymbol{R}^a \in \mathbb{R}^{|\mathcal{S}|}$.

Let $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ be a (Markov) policy and $\pi(a|s)$ indicate the probability of action a being executed in state s. Given policy π, the interaction between a learner and its environment is described as follows. The learner in state s_t uses policy $\pi(a_t|s_t)$ to determine action a_t. The learner receives reward r_{t+1} and moves to next state s_{t+1}, all of which is determined by reward function $\mathcal{R}_M(s_t, a_t)$ and transition probability $\mathcal{P}_M(s_{t+1}|s_t, a_t)$, see Fig. 2a. It is clear that the sequence of visited states $\{s_t\}_t$ is a Markov chain with transition probability $\bar{\boldsymbol{P}}^\pi$; its elements are defined as $[\bar{\boldsymbol{P}}^\pi]_{ij} = \sum_{a\in\mathcal{A}} \pi(a|s = i)[\boldsymbol{P}^a]_{ij}$.

The *state value function* $\mathcal{V}^\pi_M$ and *action value function* $\mathcal{Q}^\pi_M$ are defined using the expected discounted sum of the future reward when following policy π as follows,

$$\mathcal{V}^\pi_M(s) = \lim_{T\to\infty} \mathbb{E}^\pi_{\boldsymbol{d}^T}\Big[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)\Big| s_0 = s\Big], \quad (1)$$

$$\mathcal{Q}^\pi_M(s, a) = \lim_{T\to\infty} \mathbb{E}^\pi_{\boldsymbol{d}^T}\Big[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)\Big| s_0{=}s, a_0{=}a\Big], \quad (2)$$

where $\mathbb{E}^\pi_{\boldsymbol{d}^T}$ denotes the expectation over trajectory $\boldsymbol{d}^T = \{s_t, a_t\}_{t=0}^T$ following π. It is well known that value functions satisfy the *Bellman expectation equation*, which can be written as

$$\mathcal{V}^\pi_M(s){=}\mathbb{E}_{a\sim\pi(a|s), s'\sim\mathcal{P}_M(s'|s,a)}\big[\mathcal{R}_M(s, a){+}\gamma\mathcal{V}^\pi_M(s')\big], \quad (3)$$

$$\mathcal{Q}^\pi_M(s, a){=}\mathcal{R}_M(s, a){+}\gamma\mathbb{E}_{s'\sim\mathcal{P}_M(s'|s,a)}\big[\mathcal{V}^\pi_M(s')\big]. \quad (4)$$

Similarly, *the optimal action value function*, which is defined as $\mathcal{Q}^*_M(s, a) = \max_\pi \mathcal{Q}^\pi_M(s, a)$, satisfies the following *Bellman optimality equation*:

$$\mathcal{Q}^*_M(s, a){=}\mathcal{R}_M(s, a){+}\gamma\mathbb{E}_{s'}\big[\max_{a'} \mathcal{Q}^*_M(s', a')\big]. \quad (5)$$

Policy π^* that satisfies $\mathcal{Q}^*_M(s, a) = \mathcal{Q}^{\pi^*}_M(s, a)$ for all (s, a) is known as the *optimal policy*. These equations can be used to derive various planning algorithms such as value iteration and policy iteration, and various reinforcement leaning algorithms such as TD learning and Q learning. In a later section, we will use vector representations of the value functions; state value function $\mathcal{V}^\pi_M$ can be written as

$$\boldsymbol{V}^\pi = \bar{\boldsymbol{R}}^\pi + \gamma\bar{\boldsymbol{P}}^\pi\boldsymbol{V}^\pi \Leftrightarrow \boldsymbol{V}^\pi = \{\boldsymbol{I} - \gamma\bar{\boldsymbol{P}}^\pi\}^{-1}\bar{\boldsymbol{R}}^\pi,$$

where $[\boldsymbol{V}^\pi]_i = V^\pi_M(s{=}i)$ and $[\bar{\boldsymbol{R}}^\pi]_i = \sum_a \pi(a|s{=}i)[\boldsymbol{R}^a]_i$.

B. Option and Semi-Markov Decision Processes

In standard MDP, every action is completed in each step. In contrast, MDPs with Option (OMDPs) use the macro action called *option*; it execute actions multiple times. The formal definition of Option is given as follows.

Definition 1 (Option) [22] *Option is defined by three components, policy μ: $\mathcal{S} \times \mathcal{A} \to [0, 1]$, termination condition β: $\mathcal{S} \to [0, 1]$, and initiation set $\mathcal{I} \subseteq \mathcal{S}$, $< \mu, \beta, \mathcal{I} >$.*

Note that standard action can also be viewed as an option that always terminates right after execution. We denote a set of options as $\mathcal{O}$. In this subsection, we consider that policy π is defined as the probability of executing an option, i.e., $\pi : \mathcal{S} \times \mathcal{O} \to [0, 1]$.

The interaction of an agent and environment in OMDP is described as follows. In state s_t, option o is a execution candidate if its initiation set $\mathcal{I}$ includes s_t. The agent decides to execute option o_t following policy $\pi(o_t|s_t)$ and then conducts action a_t following the policy of the option $\mu(a_t|s_t)$. The learner then receives reward r_{t+1} and moves to the next state, s_{t+1}, as determined by reward function $\mathcal{R}_M(s_t, a_t)$ and transition probability $\mathcal{P}_M(s_{t+1}|s_t, a_t)$. In addition, the termination of the option is (randomly) determined following $\beta(s_{t+1})$. If it is not terminated, the next action is executed following the policy of running option $\mu(a_{t+1}|s_{t+1})$. If the option is terminated, the next option is determined according to policy $\pi(o_{t+1}|s_{t+1})$, see Fig. 2b. Since the time interval of decision making under policy π varies which yields the following result:

Proposition 1 [22] *OMDP is a special case of semi-MDP.*

The advantage of using OMDP is that enables us to handle time-inconsistent decision making in a manner analogous to standard MDP by redefining the reward function and transition probability.

Definition 2 (Transition Probability over Option) [22] $\mathcal{P}_O(s'|s, o) = \sum_{k=1}^\infty \rho(s', k)\gamma^{k-1}$, *where $\rho(s', k)$ is the probability that option o is terminated in state s' after k step transitions.*

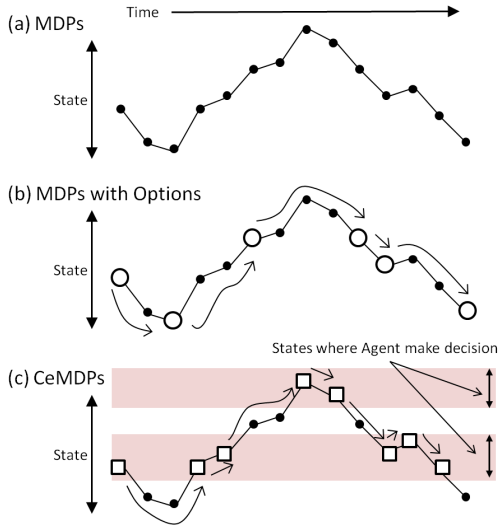


Fig. 2: Example of trajectory of (a) MDP, (b) MDP with option (OMDP) and (c) Censored MDP (CeMDP). The learner makes its decision following the learner's policy at (a) every time step, (b) random time steps when the running option is terminated and (c) random time steps when the leaner is in one of a set of known states $\mathcal{L}$.

Definition 3 (Reward Function over Option) [22] $\mathcal{R}_O(s,o) = \mathbb{E}[r_{t+1}+\gamma r_{t+2}+\cdots+\gamma^{k-1}r_{t+k}|s_t=s,o_t=o]$, *where $t+k$ is the time (random) at which option o is terminated.*

Since the above definitions show that the outcomes of an option at many different times are combined, OMDP can be seen as a multi-time model [22], [23], [24]. A later section converts CeMDP into MDP in a similar manner. By defining the value function of OMDP similar to Eq. (1)(2), the following Bellman equation for OMDP is satisfied:

$$\mathcal{V}_O^\pi(s)=\mathbb{E}_{o\sim\pi(o|s),s'\sim\mathcal{P}_O(s'|s,o)}\left[\mathcal{R}_O(s,o)+\gamma\mathcal{V}_O^\pi(s')\right], \quad (6)$$

$$\mathcal{Q}_O^*(s,o)=\mathcal{R}_O(s,o)+\gamma\mathbb{E}_{s'\sim\mathcal{P}_O(s'|s,o)}\left[\max_{o'}\mathcal{Q}_O^*(s',o')\right] \quad (7)$$

Therefore, the planning methods used for standard MDP such as value iteration and policy iteration [22] can be applied to OMDP in an analogous manner. We can also apply RL methods, e.g., Temporal Difference (TD) learning and Q-learning to OMDP. Q-learning updates the estimator of $\mathcal{Q}_O^*$, $\hat{\mathcal{Q}}_O$, at the time at which the option is terminated as follows:

$$\hat{\mathcal{Q}}_O(s_t,o_t) \leftarrow \hat{\mathcal{Q}}_O(s_t,o_t)+\eta_t\delta_t, \quad (8)$$

$$\delta_t = \{\sum_{k=1}^{k'}\gamma^{k-1}r_{t+k}\}+\gamma^{k'}\max_{o'}\hat{\mathcal{Q}}_O(s_{t+k'},o')-\hat{\mathcal{Q}}_O(s_t,o_t),$$

where η_t is the learning rate and $t+k'$ is the time at which option o_t is terminated. The difference from Q-learning in standard MDP lies in computing a (discounted) cumulative reward over k' step transition; the discounted factor is raised to the power of k'.

Remark: We use a slightly different definition of transition probability for option (Definition 2) from the original one [22]: Originally, the probability was defined as $\mathcal{P}_O(s'|s,o) = \sum_{k=1}^{\infty}\rho(s',k)\gamma^k$. This creates a notational difference from the Bellman equation, where discounted rate γ is multiplied by the value function in the r.h.s of Eq. (6). See Eq. (6) in our paper and Eq. (8) in [22].

III. Censored Markov Decision Process (CeMDP)

This section defines Censored MDP, a new MDP framework for describing the interaction between learner, external system, and environment as shown in Fig. 1. We also clarify its relationship to OMDP, semi-MDP, and MDP.

A. Definition

We define CeMDP by the septuplet $\{\mathcal{S},\mathcal{A},\mathcal{P}_M,\mathcal{R}_M,\gamma,\mathcal{E},\mu\}$, where $\mathcal{S}$, $\mathcal{A}$, $\mathcal{P}_M$, $\mathcal{R}$, γ is a set of states, a set of actions, transition probability, reward function and discount factor, respectively. $\mathcal{E}\subseteq\mathcal{S}$ is a set of states where the external system has the right to make decisions as to actions. μ is a predefined policy on $\mathcal{E}$, $\mu:\mathcal{E}\times\mathcal{A}\rightarrow[0,1]$. We also denote the set of states where the learner has the right to make decisions as $\mathcal{L}$. Note that $\mathcal{E}\cup\mathcal{L}=\mathcal{S}$, $\mathcal{E}\cap\mathcal{L}=\emptyset$.

Given policy $\pi_\ell:\mathcal{L}\times\mathcal{A}\rightarrow[0,1]$, the interaction of learner, external system and environment in CeMDP is described as follows. At each time t, when the system is in a state in $\mathcal{L}$, action a_t is determined following the learner's policy $\pi_\ell(a_t|s_t)$. If the system is in a state in $\mathcal{E}$, the action is determined by the external system's policy $\mu(a_t|s_t)$. The system consequently receives reward r_{t+1} and moves to the next state, s_{t+1}, in accordance with reward function $\mathcal{R}_M(s_t,a_t)$ and transition probability $\mathcal{P}_M(s_{t+1}|s_t,a_t)$. See Fig. 1 and Fig. 2c. Note that if $\mathcal{E}$ is an empty set, this interaction is equivalent to that of standard MDP.

In the analysis shown later, we refer to MDP where $\mathcal{E}$ and pre-defined policy μ are excluded from CeMDP as *original MDP*. A formal definition is given below:

Definition 4 (Original MDP of CeMDP) *We call MDP $\{\mathcal{S},\mathcal{A},\mathcal{P}_M,\mathcal{R}_M,\gamma\}$ the original MDP of CeMDP $\{\mathcal{S},\mathcal{A},\mathcal{P}_M,\mathcal{R}_M,\gamma,\mathcal{E},\mu\}$.*

B. CeMDP as MDP with Option (OMDP)

This subsection shows that CeMDP can be seen as a special case of OMDP [22]. It can be shown that OMDP can reproduce the interaction of CeMDP by using a specific definition of option.

Proposition 2 *CeMDP is a special case of OMDP.*

Proof: For each action a, we define Option $o_a =< \nu_a,\beta_a,\mathcal{I}_a >$, where initiation set $\mathcal{I}_a$ is $\mathcal{L}$, policy ν_a is $\nu_a(a'|s)=1.0$ if $a=a'$ and $s\in\mathcal{L}$, $\nu_a(a'|s)=0.0$ if $a\neq a'$ and $s\in\mathcal{L}$, and $\nu_a(a'|s)=\mu(a'|s)$ otherwise. The termination condition β_a is given by $\beta_a(s)=1.0$ if $s\in\mathcal{L}$, otherwise $\beta_a(s)=0.0$. This makes the interaction of agent and environment equivalent to that of CeMDP. ■

From Proposition 1 and 2, the following corollary is immediately obtained.

Corollary 1 *CeMDP is a special case of semi-MDP.*

Proposition 2 also means that we can treat CeMDP as OMDP. Therefore, various algorithms developed for OMDP can also be used for CeMDP as shown by the example algorithms in Sec. IV. First, the theoretical relationship between CeMDP and original MDP is examined in more detail.

C. Converting CeMDP to MDP on $\mathcal{L}$

This subsection is devoted to elucidating the conversion of CeMDP into MDP. As shown in Fig. 1, the learner does not need to make decisions if the state is in $\mathcal{E}$. Thus, we investigate the conversion of CeMDP to the one whose state space is given by $\mathcal{E}$. This can be done by redefining the environment by integrating original environment and the external system. Figure 3 shows the interaction yielded by such an environment. Based on the definition of transition probability and reward in OMDP (Def. 2 and 3), we define the following transition probability and reward, both of which correspond to the redefined environment of CeMDP.

Definition 5 *Let us define transition probability* $\mathcal{P}_C : \mathcal{L} \times \mathcal{L} \times \mathcal{A} \to [0,1]$ *where* $\mathcal{P}_C(s'|s,a)$ *represents the discounted sum of the probability that state* s' *is visited by executing action* a *in state* s *without visiting any states in* $\mathcal{L}$ *before state* s' *is visited, i.e.,* $\mathcal{P}_C(s'|s,a) = \sum_{k=1}^{\infty} \lambda(s',k)\gamma^{k-1}$, *where* $\lambda(s',k)$ *is the probability that state* s' *is visited at time* $t+k$ *when action* a *is executed in state* s *at time* t *without visiting any state in* $\mathcal{L}$ *before state* s' *is visited.*

Definition 6 *Let us define reward function* $\mathcal{R}_C : \mathcal{L} \times \mathcal{A} \to \mathbb{R}$ *where* $\mathcal{R}_C(s,a)$ *represents the expected discount sum of the reward prior to any state in* $\mathcal{L}$ *being visited by executing action* a *in state* s, *i.e.,* $\mathcal{R}_C(s,a) = \mathbb{E}[r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{k-1} r_{t+k} | s_t = s, a_t = a]$, *where* $t+k$ *is the first time that the agent visits some state in* $\mathcal{L}$ *after time* t.

Remark: Rigorously speaking, the above (action) transition probability and reward function are for option o_a (See Proof of Prop. 2). Since the option and action have a one to one correspondence, hereinafter we use this notation.

Without loss of generality, from now on, we consider that the states are re-ordered so that the matrix representation of the transition probability, $\boldsymbol{P}^a$, and its expectation w.r.t. a following $\pi = (\pi_\ell, \pi_e)$, $\bar{\boldsymbol{P}}^\pi$ can be written in the following block matrix form:

$$\boldsymbol{P}^a = \begin{array}{c} \\ \mathcal{L} \\ \mathcal{E} \end{array} \begin{array}{c} \begin{array}{cc} \mathcal{L} & \mathcal{E} \end{array} \\ \begin{pmatrix} \boldsymbol{P}^a_{\ell\ell} & \boldsymbol{P}^a_{\ell e} \\ \boldsymbol{P}^a_{e\ell} & \boldsymbol{P}^a_{ee} \end{pmatrix} \end{array}, \quad \bar{\boldsymbol{P}}^\pi = \begin{array}{c} \\ \mathcal{L} \\ \mathcal{E} \end{array} \begin{array}{c} \begin{array}{cc} \mathcal{L} & \mathcal{E} \end{array} \\ \begin{pmatrix} \bar{\boldsymbol{P}}^{\pi_\ell}_{\ell\ell} & \bar{\boldsymbol{P}}^{\pi_\ell}_{\ell e} \\ \bar{\boldsymbol{P}}^{\pi_e}_{e\ell} & \bar{\boldsymbol{P}}^{\pi_e}_{ee} \end{pmatrix} \end{array}, \tag{9}$$

We also denote the vector made by extracting the values at states in $\mathcal{L}$ ($\mathcal{E}$) as the vector with lower script ℓ (e), e.g., $\boldsymbol{R}^a = (\boldsymbol{R}^a_\ell, \boldsymbol{R}^a_e)$, $\bar{\boldsymbol{R}}^\pi = (\bar{\boldsymbol{R}}^\pi_\ell, \bar{\boldsymbol{R}}^\pi_e)$. Then, the following theorem holds.

Proposition 3 (Analytic Form of Transition Probability) *Matrix representation of transition probability* $\mathcal{P}_C$ *given action* a, $\boldsymbol{P}^a_C$, *is given by* $\boldsymbol{P}^a_C = \boldsymbol{P}^a_{\ell\ell} + \gamma \boldsymbol{P}^a_{\ell e}(\boldsymbol{I} - \gamma \bar{\boldsymbol{P}}^\mu_{ee})^{-1} \bar{\boldsymbol{P}}^\mu_{e\ell}$.

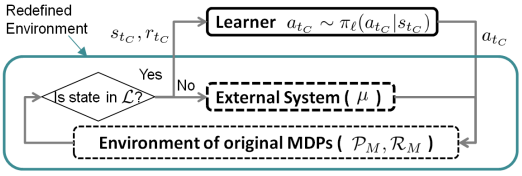


Fig. 3: Interaction of learner and (redefined) environment where external system is integrated in CeMDP. (Cf. Fig. 1)

Proposition 4 (Analytic Form of Reward Function) *Vector representation of reward function* $\mathcal{R}_C$ *given action* a, $\boldsymbol{R}^a_C$, *is given by* $\boldsymbol{R}^a_C = \boldsymbol{R}^a_\ell + \gamma \boldsymbol{P}^a_{\ell e}(\boldsymbol{I} - \gamma \bar{\boldsymbol{P}}^\mu_{ee})^{-1} \bar{\boldsymbol{R}}^\mu_e$.

For deriving the above theorems, we introduce the following two quantities; these are also used for algorithm construction.

Definition 7 (Discounted Absorbing Probability) *Let* $\mathcal{T}^\mu : \mathcal{E} \times \mathcal{L} \to [0,1]$ *be the (discounted) absorbing probability which indicates that state* $s' \in \mathcal{L}$ *is reached from state* $s \in \mathcal{E}$ *without visiting any state in* $\mathcal{L}$ *on the way:*

$$\mathcal{T}^\mu(s,s') = \sum\nolimits_{k=1}^{\infty} p_A(s'|s;k)\gamma^{k-1}, \tag{10}$$

where $p_A(s',s;k)$ *is the probability that state* $s' \in \mathcal{L}$ *is visited from state* $s \in \mathcal{E}$ *after* k *step transitions without visiting any state in* $\mathcal{L}$ *on the way.*

Definition 8 (Discounted Absorbing Reward) *Let* $\mathcal{U} : \mathcal{E} \to \mathbb{R}$ *be the (discounted) sum of rewards until some state in* $\mathcal{L}$ *is visited from state* $s \in \mathcal{E}$ *without visiting any state in* $\mathcal{L}$ *on the way:*

$$\mathcal{U}^\mu(s) = \mathbb{E}[r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{k-1} r_{t+k} | s_t = s], \tag{11}$$

where $t+k$ *is the first time that the agent visits some state in* $\mathcal{L}$ *after time* t.

$\mathcal{T}^\mu$ and $\mathcal{U}^\mu$ satisfy the following recursive equations.

Lemma 1 *The discounted absorbing probability (Eq.* (10)*) satisfies the following recursive equation:*

$$\begin{aligned} &\mathcal{T}^\mu(s'|s) \\ &= \mu(a|s)\Big[\mathcal{P}_M(s'|s,a) + \gamma \sum\nolimits_{s'' \in \mathcal{E}} \mathcal{P}_M(s''|s,a)\mathcal{T}^\mu(s'|s'')\Big]. \end{aligned} \tag{12}$$

Lemma 2 *The discounted absorbing reward (Eq.* (11)*) satisfies the following recursive equation:*

$$\mathcal{U}^\mu(s) = \pi(a|s)\Big[\mathcal{R}_M(s,a) + \gamma \sum_{s' \in \mathcal{E}} \mathcal{P}_M(s'|s,a)\mathcal{U}^\mu(s')\Big] \tag{13}$$

The proofs of Lemma 1 and 2 are shown in the Appendix. Note that Eq. (12) and (13) can be written in the following form [1]:

$$\begin{aligned} \boldsymbol{T}^\mu &= \bar{\boldsymbol{P}}^\mu_{e\ell} + \gamma \bar{\boldsymbol{P}}^\mu_{ee}\boldsymbol{T}^\mu \Leftrightarrow \boldsymbol{T}^\mu = (\boldsymbol{I} - \gamma \bar{\boldsymbol{P}}^\mu_{ee})^{-1} \bar{\boldsymbol{P}}^\mu_{e\ell} \\ \boldsymbol{U}^\mu &= \bar{\boldsymbol{R}}^\mu_e + \gamma \bar{\boldsymbol{P}}^\mu_{ee}\boldsymbol{U}^\mu \Leftrightarrow \boldsymbol{U}^\mu = (\boldsymbol{I} - \gamma \bar{\boldsymbol{P}}^\mu_{ee})^{-1} \bar{\boldsymbol{R}}^\mu_e \end{aligned}$$

[1]The existence of inverse matrices are assured.

Proof: (Proposition 3) Since $\mathcal{P}_C(s'|s,a)$ can be decomposed into the probability of moving directly from s to s' and the probability that the movement passes through $s'' \in \mathcal{E}$,

$$\mathcal{P}_C(s'|s,a) = \lambda(s'|s;1) + \sum\nolimits_{k=2}^{\infty} \lambda(s'|s;k)\gamma^{k-1} \\ = \mathcal{P}_M(s'|s,a) + \gamma \sum\nolimits_{s''\in\mathcal{E}} \mathcal{P}_M(s''|s,a)\mathcal{T}^{\mu}(s'|s'')\Big].$$

Its vector representation is written as

$$\boldsymbol{P}_C^a = \boldsymbol{P}_{\ell\ell}^a + \gamma \boldsymbol{P}_{\ell e}^a \boldsymbol{T}^{\mu} = \boldsymbol{P}_{\ell\ell}^a + \gamma \boldsymbol{P}_{\ell e}^a (\boldsymbol{I} - \gamma \bar{\boldsymbol{P}}_{ee}^{\mu})^{-1} \bar{\boldsymbol{P}}_{e\ell}^{\mu}. \quad (14)$$

■

Proof: (Proposition 4) Since $\mathcal{R}_C(s,a)$ can be decomposed into the reward obtained when directly moving from s to s' and the reward received when going through $s'' \in \mathcal{E}$,

$$\mathcal{R}_C(s,a) = \sum\nolimits_{s'\in\mathcal{L}} \mathcal{P}_M(s'|s,a)\mathcal{R}_M(s,a) \\ + \sum\nolimits_{s''\in\mathcal{E}} \mathcal{P}_M(s''|s,a)\big[\mathcal{R}_M(s,a) + \mathcal{U}^{\mu}(s'')\big] \\ = \mathcal{R}_M(s,a) + \sum\nolimits_{s''\in\mathcal{E}} \mathcal{P}_M(s''|s,a)\mathcal{U}^{\mu}(s'').$$

Its vector representation is written as

$$\boldsymbol{R}_C^a = \boldsymbol{R}_{\ell}^a + \gamma \boldsymbol{P}_{\ell e}^a \boldsymbol{U}^{\mu} = \boldsymbol{R}_{\ell}^a + \gamma \boldsymbol{P}_{\ell e}^a (\boldsymbol{I} - \gamma \bar{\boldsymbol{P}}_{ee}^{\mu})^{-1} \bar{\boldsymbol{R}}_e^{\mu}. \quad (15)$$

■

Based on Proposition 3 and 4, we define the MDP representation of CeMDP as follows:

Definition 9 (MDP representation of CeMDP) *We define the MDP representation of CeMDP* $\{\mathcal{S},\mathcal{A},\mathcal{P}_M,\mathcal{R}_M,\gamma,\mathcal{E},\mu\}$ *as* $\{\mathcal{L},\mathcal{A},\mathcal{P}_C,\mathcal{R}_C,\gamma\}$ *where the transition probability* $\mathcal{P}_C = \{\boldsymbol{P}_C^a\}_{a\in\mathcal{A}}$ *and reward function* $\mathcal{R}_C = \{\boldsymbol{R}_C^a\}$ *are given by Eq.* (14) *and* (15)*, respectively.*

We again emphasize that this representation allows us to reduce the state space from $\mathcal{S}$ to $\mathcal{L}$.

D. Equivalence of Value Function

We define the value function for CeMDP by using the MDP representation of CeMDP following Eq. (1)(2). It can be shown that it satisfies the following Bellman equation:

$$\mathcal{V}_C^{\pi_\ell}(s) = \mathbb{E}_{a\sim\pi_\ell(a|s), s'\sim\mathcal{P}_C(s'|s,a)}\big[\mathcal{R}_C(s,a) + \gamma \mathcal{V}_C^{\pi_\ell}(s')\big], \\ \mathcal{Q}_C^*(s,a) = \mathcal{R}_C(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}_C(s'|s,a)}\big[\max\nolimits_{a'} \mathcal{Q}_C^*(s',a')\big],$$

where π_ℓ is a policy on $\mathcal{L}$. We denote the ($|\mathcal{L}|$ dimensional) vector representation of the above value function as $\boldsymbol{V}_C^{\pi_\ell}$

$$\boldsymbol{V}_C^{\pi_\ell} = \bar{\boldsymbol{R}}_C^{\pi_\ell} + \gamma \bar{\boldsymbol{P}}_C^{\pi_\ell} \boldsymbol{V}_C^{\pi_\ell} \Leftrightarrow \boldsymbol{V}_C^{\pi_\ell} = \{\boldsymbol{I} - \gamma \bar{\boldsymbol{P}}_C^{\pi_\ell}\}^{-1} \bar{\boldsymbol{R}}_C^{\pi_\ell}, \quad (16)$$

where $[\boldsymbol{V}_C^{\pi_\ell}]_i = V_C^{\pi_\ell}(s=i)$ and $[\bar{\boldsymbol{R}}_C^{\pi_\ell}]_i = \sum_a \pi_\ell(a|s=i)[\boldsymbol{R}_C^a]_i$. We also denote (π_ℓ,μ) as a policy on $\mathcal{S}$ which is a concatenation of policy π_ℓ for states in $\mathcal{L}$ and pre-defined policy μ for states not in $\mathcal{L}$.

We can show the equivalence of the value function of CeMDP and that of original MDP. This result imply the validity of MDP representation of CeMDP (Definition 9).

Proposition 5 (Equivalence of Value Function) *Value function of original MDP* $\{\mathcal{S},\mathcal{A},\mathcal{P}_M,\mathcal{R}_M,\gamma\}$ *with policy* $\pi = (\pi_o,\pi_e)$ *equals that of CeMDP* $\{\mathcal{S},\mathcal{A},\mathcal{P}_M,\mathcal{R}_M,\gamma,\mathcal{E},\pi_e\}$ *with policy* π_o *for all states in* $\mathcal{L}$*, i.e.,* $V_M^{\pi}(s) = V_C^{\pi_o}(s)$ *for all* $s \in \mathcal{L}$*.*

Proof: The vector representation of value function of original MDP, $\boldsymbol{V}^{\pi} = (\boldsymbol{V}_{\ell}^{\pi}, \boldsymbol{V}_e^{\pi})$, can be written in the following block matrix form:

$$\begin{pmatrix} \boldsymbol{V}_{\ell}^{\pi} \\ \boldsymbol{V}_e^{\pi} \end{pmatrix} = \begin{pmatrix} \boldsymbol{I} - \gamma \bar{\boldsymbol{P}}_{\ell\ell}^{\pi_\ell} & -\gamma \bar{\boldsymbol{P}}_{\ell e}^{\pi_\ell} \\ -\gamma \bar{\boldsymbol{P}}_{e\ell}^{\pi_e} & \boldsymbol{I} - \gamma \bar{\boldsymbol{P}}_{ee}^{\pi_e} \end{pmatrix}^{-1} \begin{pmatrix} \bar{\boldsymbol{R}}_{\ell}^{\pi_\ell} \\ \bar{\boldsymbol{R}}_e^{\pi_e} \end{pmatrix} \quad (17) \\ = \begin{pmatrix} (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C})^{-1} \bar{\boldsymbol{R}}_{\ell}^{\pi_\ell} + \boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1} \bar{\boldsymbol{R}}_e^{\pi_e} \\ (\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} \bar{\boldsymbol{R}}_{\ell}^{\pi_\ell} + (\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1} \bar{\boldsymbol{R}}_e^{\pi_e} \end{pmatrix},$$

where $\boldsymbol{A} = \boldsymbol{I} - \gamma \bar{\boldsymbol{P}}_{\ell\ell}^{\pi_\ell}$, $\boldsymbol{B} = \gamma \bar{\boldsymbol{P}}_{\ell e}^{\pi_\ell}$, $\boldsymbol{C} = \gamma \bar{\boldsymbol{P}}_{e\ell}^{\pi_e}$ and $\boldsymbol{D} = \boldsymbol{I} - \gamma \bar{\boldsymbol{P}}_{ee}^{\pi_e}$. From Eq. (14)(15)(16), the value function and reward function of CeMDP are written as

$$\boldsymbol{V}_C^{\pi_\ell} = \{\boldsymbol{I} - \gamma \bar{\boldsymbol{P}}_C^{\pi_\ell}\}^{-1} \bar{\boldsymbol{R}}_C^{\pi_\ell} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C})^{-1} \bar{\boldsymbol{R}}_C^{\pi_\ell}, \quad (18) \\ \bar{\boldsymbol{R}}_C^{\pi_\ell} = \bar{\boldsymbol{R}}_{\ell}^{\pi_\ell} + \gamma \bar{\boldsymbol{P}}_{\ell e}^{\pi_\ell} (\boldsymbol{I} - \gamma \bar{\boldsymbol{P}}_{ee}^{\pi_e})^{-1} \bar{\boldsymbol{R}}_e^{\pi_e} = \bar{\boldsymbol{R}}_{\ell}^{\pi_\ell} + \boldsymbol{B}\boldsymbol{D}^{-1} \bar{\boldsymbol{R}}_e^{\pi_e}. \quad (19)$$

From Eq. (17)(18), $\boldsymbol{V}_{\ell}^{\pi} = \boldsymbol{V}_C^{\pi_\ell}$ holds when

$$\bar{\boldsymbol{R}}_C^{\pi_\ell} = \bar{\boldsymbol{R}}_{\ell}^{\pi_\ell} + (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C})\boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1} \bar{\boldsymbol{R}}_e^{\pi_e}.$$

We can show this equation is equivalent to Eq. (19):

$$\bar{\boldsymbol{R}}_C^{\pi_\ell} = \bar{\boldsymbol{R}}_{\ell}^{\pi_\ell} + \boldsymbol{B}(\boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})(\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1} \bar{\boldsymbol{R}}_e^{\pi_e} \\ \Leftrightarrow \bar{\boldsymbol{R}}_C^{\pi_\ell} = \bar{\boldsymbol{R}}_{\ell}^{\pi_\ell} + \boldsymbol{B}\boldsymbol{D}^{-1} \bar{\boldsymbol{R}}_e^{\pi_e}.$$

Thus, $\boldsymbol{V}_{\ell}^{\pi} = \boldsymbol{V}_C^{\pi_\ell}$ holds. ■

Proposition 5 holds for arbitrary policy, and the following corollaries are immediately obtained.

Corollary 2 (Equivalence of Optimal Policy) *Let* $\pi^* = (\pi_{\ell}^*, \pi_e^*)$ *be the optimal policy of original MDP* $\{\mathcal{S},\mathcal{A},\mathcal{P}_M,\mathcal{R}_M,\gamma\}$*. Then,* π_{ℓ}^* *is the optimal policy of CeMDP* $\{\mathcal{S},\mathcal{A},\mathcal{P}_M,\mathcal{R}_M,\gamma,\mathcal{E},\pi_e^*\}$*.*

Proof: From Proposition 5, $V^{\pi^*}(s) = V_C^{\pi_{\ell}^*}(s)$ holds. Suppose policy π_{ℓ}' that satisfies $V_C^{\pi_{\ell}^*}(s) \le V_C^{\pi_{\ell}'}(s)$ for some $s \in \mathcal{L}$ exists. This yields $V^{\pi^*}(s) \le V^{(\pi_{\ell}', \pi_e^*)}(s)$, which contradicts the fact that π^* is the optimal policy of MDP. Thus, π_{ℓ}^* must be an optimal policy of CeMDP. ■

IV. Algorithm for CeMDP

We design CeMDP algorithms based on the analysis provided in the previous section.

A. Reinforcement Learning Algorithm

It is shown that CeMDP is a special case of OMDP by Proposition 2. This means that RL algorithm for OMDP can be used for CeMDP. If *a prior* knowledge about the environment is unavailable, we can use e.g., TD-learning and Q-learning with some slight modification. Here we examine CeMDP Q-Learning, which we refer to as CenQ-Learning.

CeMDP Q-Learning (CenQ-Learning): The update equation of estimator $\hat{\mathcal{Q}}_C$ for $\mathcal{Q}_C^*$ is given by

$$\hat{\mathcal{Q}}_C(s_t,a_t) \leftarrow \hat{\mathcal{Q}}_C(s_t,a_t) + \eta_t \delta_t, \quad (20)$$

$$\delta_t = \Big\{\sum_{k=1}^{k'} \gamma^{k-1} r_{t+k}\Big\} + \gamma^{k'} \max_a \hat{\mathcal{Q}}_C(s_{t+k'},a) - \hat{\mathcal{Q}}_C(s_t),$$

Algorithm 1 CeMDP Q-learning (CenQ-Learning)

Input: μ: pre-defined policy, π: policy, α_t: learning rate [2]
Output: $\hat{\mathcal{Q}}(s,a)$: estimator of optimal value function

1: % Step1: Initialization
2: Set $\mathcal{Q}(s,a)=0$ for all $s\in\mathcal{E}, s'\in\mathcal{L}$ [3]
3: Set time of $t=0$ and that of CeMDP $t_C=-1$.
4: Set $k=0$ and $R_{\text{kstep}}=0$. % for tracking the number of steps and return until visiting state in $\mathcal{L}$
5: **repeat**
6: % Step2: Interaction with Environment
7: **if** $s_t\in\mathcal{L}$ **then**
8: Set $t_C\leftarrow t_C+1$, $k\leftarrow 0$, $R_{\text{kstep}}\leftarrow 0$
9: Select action a_t following $\pi(a_t|s_t)$ and execute.
10: Set $s_{\text{pre}}=s_t$ and $a_{\text{pre}}=a_t$.
11: **else**
12: Select action a_t following $\mu(a_t|s_t)$ and execute.
13: **end if**
14: Observe r_{t+1} and moves to state s_{t+1}.
15: Set $k\leftarrow k+1$
16: Set $R_{\text{kstep}}\leftarrow R_{\text{kstep}}+r_{t+1}\gamma^{k-1}$.
17: % Step3: Learning
18: **if** $s_{t+1}\in\mathcal{L}$ and $t_C\geq 0$ **then**
19: Compute $\delta_t=R_{\text{kstep}}+\gamma^k\max_a\hat{\mathcal{Q}}_C(s_{t+1},a)-\hat{\mathcal{Q}}_C(s_{\text{pre}},a_{\text{pre}})$
20: Update $\hat{\mathcal{Q}}_C(s_{\text{pre}},a_{\text{pre}})\leftarrow\hat{\mathcal{Q}}_C(s_{\text{pre}},a_{\text{pre}})+\eta_t\delta_t$.
21: **end if**
22: Set $t\leftarrow t+1$
23: **until** A stopping condition is met

where η_t is the learning rate and $t+k'$ is the first time that the agent visits some state in $\mathcal{L}$ after time t. Algorithm 1 is the pseudo code. At step 2, the next action is decided by the learner or external system depending on state s_t. If $s_t\in\mathcal{L}$, record s_t, a_t and initialize the number of steps k and sum of rewards R_{kstep}. When reward r_t and next states s_t are observed, update k and R_{kstep}. At step 3, the value function is updated using k and R_{kstep} when s_{t+1} is in $\mathcal{L}$. Since CenQ-Learning can be seen as a variant of Q-learning for OMDP, it converges to the optimal value function under conditions similar to those for Q-learning [22], [25].

B. Planning Algorithm

The use of the MDP representation of CeMDP directly yields a two stage planning algorithm that computes $\mathcal{T}^\mu$ and $\mathcal{U}^\mu$ and then computes $\mathcal{V}_C$ via a MDP planning algorithm [4].

Computing Absorbing Probability and Reward: We can obtain $\mathcal{T}^\mu$ and $\mathcal{U}^\mu$ by a value-iteration-like algorithm since they satisfy the recursive equation Eq. (12) and (13). Algorithm 2 and 3 show the pseudo code. The convergence can be proven although we omit the proof for lack of space.

Value Iteration: Once $\mathcal{T}^\mu$ and $\mathcal{U}^\mu$ are obtained, reward function $\mathcal{R}_C$ and transition probability $\mathcal{R}_C$ can be computed using Eq. (14) and (15). Therefore, we can use MDP planning algorithms such as value iteration and policy iteration. Algorithm 4 employs the value iteration. Note that since it uses MDP representation of CeMDP, the state space is reduced from $\mathcal{S}$ to $\mathcal{L}$. This two-stage approach is useful when, for example, reward function on $\mathcal{L}$, $\mathcal{R}^a_\ell$, must be designed by trial and error. Since $\mathcal{T}^\mu$ and $\mathcal{U}^\mu$ do not depend on $\boldsymbol{R}^a_\ell$, they can be reused even if $\boldsymbol{R}^a_\ell$ changes. The value function is easier to obtain as state space $\mathcal{L}$ is smaller.

V. DISCUSSION

Insight For Safe RL with External System: Our analysis is useful for designing a safe RL system that uses an external system (e.g., human). As shown in CenQ-Learning, although the number of steps and cumulative rewards need to be recorded (Line 15-16) for updating the value function even if an external system makes the decision, we do not need to record the action executed by the external system. We consider this to be beneficial in developing systems with less effort since collecting a complete behavior of external system such as human may be infeasible.

Comparison with OMDP: We show that CeMDP is a special case of OMDP (Proposition 2). Moreover, we clarify that CeMDP can be seen as MDP with redefined environment whose state space is reduced and whose transition probability and reward function are given in Proposition 3 and 4. This allows us to develop the planning algorithm for CeMDP. We consider these finding also provides a deeper insights of OMDP.

Relation to Censored Markov Chain (CMC): We can also show that CeMDP is strongly related to CMC. CMC is constructed from the original Markov Chain (MC) $\{X_t; t=0,1,2,\cdots\}$ and a set of observable states $\mathcal{W}$. Formally, CMC is a stochastic process $\{X^c_t; t=0,1,2,\cdots\}$; the state at time t is the state of the MC at time σ_t which is the time of the t-th visit to the observable states $\mathcal{W}$, i.e., $X^c_t:=X_{\sigma_t}$ [6]. Intuitively, CMC is the MC watched only when it uses states in $\mathcal{W}$. It can be shown that CMC itself is also MC:

Proposition 6 (e.g., Lemma 6-6 [28]) *CMC constructed from the MC with transition probability* $\boldsymbol{P}=(\boldsymbol{P}_{ww},\boldsymbol{P}_{wn};\boldsymbol{P}_{nw},\boldsymbol{P}_{nn})$ *and observed states* $\mathcal{W}$ *is the MC with transition probability* $\boldsymbol{P}_{\text{CMC}}=\boldsymbol{P}_{ww}+\boldsymbol{P}_{wn}(\boldsymbol{I}-\boldsymbol{P}_{nn})^{-1}\boldsymbol{P}_{nw}$.

Therefore, CMC constructed from MC with transition probability $(\boldsymbol{P}^a_{\ell\ell},\boldsymbol{P}^a_{\ell e};\bar{\boldsymbol{P}}^\mu_{e\ell},\bar{\boldsymbol{P}}^\mu_{ee})$ and observed states $\mathcal{L}$ is given by $\boldsymbol{P}_{\text{CMC}}=\boldsymbol{P}^a_{\ell\ell}+\boldsymbol{P}^a_{\ell e}(\boldsymbol{I}-\bar{\boldsymbol{P}}^\mu_{ee})^{-1}\bar{\boldsymbol{P}}^\mu_{e\ell}$, which is equivalent to Proposition 3 with $\gamma=1.0$. We can see Proposition 3 is the discounted variant of this proposition.

[2] In a later experiment, we use ϵ-*greedy policy* $\pi(a|s;\hat{Q},\epsilon_t)$, which is defined as $\pi(a|s,\hat{Q},\epsilon_t)=1.0-\epsilon_t+\epsilon_t/|\mathcal{A}|$ if $a=\arg\max_{a'}\hat{Q}(s,a')$, and $\pi_g(a|s,\hat{Q},\epsilon_t)=\epsilon_t/|\mathcal{A}|$ otherwise. Parameter ϵ_t and learning rate η_t are set to $100/(t_C+200)$.

[3] We can use other initialization approaches such as *optimisitic initialization* [2].

[4] Although we can develop the planning algorithm based on that for MDP by fixing the policy on $\mathcal{E}$ to μ, we omit it for lack of space.

[6] [26], [27] state that CMC was constructed by Paul Lévy [18], [19], [20].

Algorithm 2 Iterative Absorbing Probability Computation

Input: $\mathcal{P}_M, \mathcal{R}_M, \gamma, \mathcal{E}, \mu$
Output: $\mathcal{T}(s,s')$: estimator of $\mathcal{T}^\mu(s,s')$
1: Initialization: Set $\mathcal{T}(s'|s)=0$ for all $s\in\mathcal{E}, s'\in\mathcal{L}$
2: **repeat**
3: $\quad \mathcal{T}(s'|s) \leftarrow \sum_a \mu(a|s)\Big[\mathcal{P}_M(s'|s,a) + \gamma\sum_{s''}\mathcal{P}_M(s''|s,a)\mathcal{T}(s'|s'')\Big]$
4: **until** A stopping condition is met

Algorithm 3 Iterative Absorbing Reward Computation

Input: $\mathcal{P}_M, \mathcal{R}_M, \gamma, \mathcal{E}, \mu$
Output: $\mathcal{U}(s)$: estimator of $\mathcal{U}^\mu(s)$
1: Initialization: Set $\mathcal{U}(s)=0$ for all $s\in\mathcal{E}$
2: **repeat**
3: $\quad \mathcal{U}(s) \leftarrow \sum_a \mu(a|s)\Big[\mathcal{R}_M(s,a)+\gamma\sum_{s'}\mathcal{P}_M(s'|s,a)\mathcal{U}(s')\Big]$
4: **until** A stopping condition is met

Algorithm 4 (Two Stage) Value Iteration for CeMDP

Input: $\mathcal{P}_M, \mathcal{R}_M, \gamma, \mathcal{E}, \mu$
Output: $\mathcal{V}_C(s)$: estimator of $\mathcal{V}_C^{\pi_\ell^*}(s)$
1: % Compute Transition Prob. and Reward Function
2: Set $\mathcal{P}_C(s'|s,a)$ following Eq. (14) for all $s,s'\in\mathcal{L}, a$
3: Set $\mathcal{R}_C(s,a)$ following Eq. (15) for all $s\in\mathcal{L}, a$
4: % Value Iteration
5: Initialization: Set $V_C(s)=0$ for all s
6: **repeat**
7: $\quad V_C(s) \leftarrow \max_a\{\mathcal{R}_C(s,a)+\gamma\sum_{s'}\mathcal{P}_C(s'|s,a)V_C(s')\}$
8: **until** A stopping condition is met

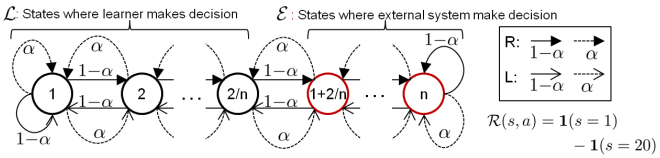


Fig. 4: n states chainwalk. $\mathbf{1}(\cdot)$ is the indicator function.

VI. EXPERIMENT

We conduct a numerical experiment to confirm the validity of our theoretical results and algorithm constructed. The experiment uses the n state chain walk shown in Fig. 4 ($n = 4, 10$). The state space and set of action are defined as $\{1,\cdots,n\}$ and $\{\mathsf{L},\mathsf{R}\}$, respectively. By action R, it moves to the right state with probability $(1-\alpha)$ and to the left state with probability α. Transition by action L is in reverse. We set $\alpha = 0.2$. The learner receives the reward of $+1.0$ at state 1 and -1.0 at state n. γ was set to 0.95. We also set the left half states $\{1,\cdots,n/2\}$ as $\mathcal{L}$ and use pre-defined policy μ, which chooses action L with probability 1, designed for avoiding the rightmost state with negative reward[7].

Absorbing Probability and Reward Computation: We first confirm the convergence of Algorithm 2 and 3. Figure 5 shows the (estimated) values of $\mathcal{T}^\mu$ and $\mathcal{U}^\mu$ at each iteration of the algorithm. It can be confirmed that both quantities converge in approximately 20 iterations. This confirms that the Algorithm 2 and 3 converge.

Value Iteration for CeMDP: We now run Algorithm 4 using the $\mathcal{T}^\mu$ and $\mathcal{U}^\mu$ computed above. Figure 6 shows the (estimated) value of $\mathcal{V}_C^*$ at each iteration. The dashed lines correspond to the optimal value function of original MDP, $\mathcal{V}_M^*$, which is computed by the value iteration algorithm for original MDP. It is confirmed that $\mathcal{V}_C$ converges to the optimal value function. This validates Corollary 2.

CenQ-Learning: Figure 7(a) shows the convergence of value function by CenQ-learning. we can see that the value converges to the true optimal value. This validates the effectiveness of CenQ-learning. We also compare the cumulative rewards obtained by CenQ and Q-learning. Figure 7(b) shows that CenQ outperforms (standard) Q-learning. Thanks to the help of the external system, CenQ-learning accumulates more rewards than Q-learning.

[7]This μ obviously corresponds to π_e^* which a part of the optimal policy of original MDP $\pi^* = (\pi_\ell^*, \pi_e^*)$.

VII. CONCLUSIONS AND FUTURE WORKS

In this study, we defined CeMDP, a framework describing the interaction between agents, external systems and environment, for safe RL; the relationships of CeMDP to semi-MDP, OMDP, and standard MDP were clarified. We showed that CeMDP is a special case of OMDP and that CeMDP can be cast as MDP variant by redefining the environment. We also constructed planning and reinforcement learning algorithm for CeMDP use and confirmed its effectiveness by numerical experiments. Remaining future work is to consider incrementally changing the pre-defined policy of the external system over time. Another promising direction is to extend our analysis to continuous time and space problems.

APPENDIX

Proof: (Lemma 1) Since $\mathcal{T}^\mu(s'|s)$ can be decomposed into the probability of direct movement from s to s' and the probability of passing through $s''\in\mathcal{E}$,

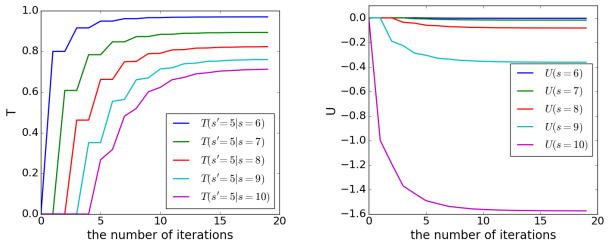
$$\begin{aligned}\mathcal{T}^\mu(s'|s) &= p_A(s'|s;1) + \sum_{k=2}^\infty p_A(s'|s;k)\gamma^{k-1}\\ &= \sum_a \mu(a|s)\Big[\mathcal{P}_M(s'|s,a)\\ &\quad + \gamma\sum_{s''\in\mathcal{E}}\mathcal{P}_M(s''|s,a)\sum_{k=1}^\infty p_A(s'|s'';k)\gamma^{k-1}\Big]\\ &= \sum_a \mu(a|s)\Big[\mathcal{P}_M(s'|s,a) + \gamma\sum_{s''\in\mathcal{E}}\mathcal{P}_M(s''|s,a)\mathcal{T}(s'|s'')\Big].\end{aligned}$$

■

Proof: (Lemma 2) Since $\mathcal{U}^\mu(s'|s)$ can be decomposed into the probability of direct movement from s to s' and the probability of passing through $s''\in\mathcal{E}$,

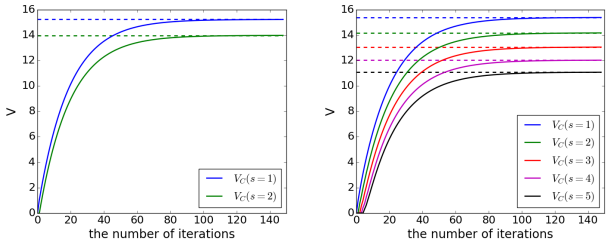
$$\begin{aligned}\mathcal{U}^\mu(s) &= \sum_{a,s'\in\mathcal{L}}\mu(a|s)\mathcal{P}_M(s'|s,a)\big[\mathcal{R}_M(s,a)\big]\\ &\quad + \sum_{a,s''\in\mathcal{E}}\mu(a|s)\mathcal{P}_M(s''|s,a)\Big\{\mathcal{R}_M(s,a)\\ &\quad + \gamma\mathbb{E}\big[\sum_{k=1}^{k'}\gamma^{k-1}r_{t+k}\big|s_t=s''\big]\Big\}\\ &= \sum_a \mu(a|s)\Big[\mathcal{R}_M(s,a) + \gamma\sum_{s''\in\mathcal{E}}\mathcal{P}_M(s''|s,a)\mathcal{U}(s'')\Big].\end{aligned}$$

■



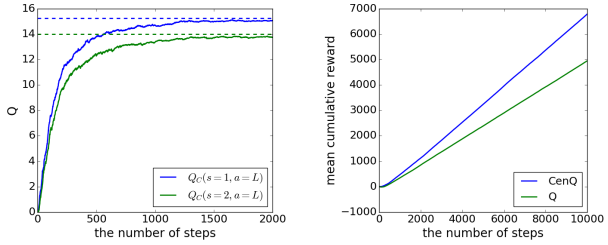
(a) T (b) U

Fig. 5: Convergence of (a) absorbing probability T^{μ} and (b) reward U^{μ} computation for $n = 10$ state chain walk



(a) $n = 4$ (b) $n = 10$

Fig. 6: Convergence of value iteration on MDP representation of CeMDP by Algorithm 4 for (a) $n = 4$ and (b) $n = 10$ state chain walk: the dotted lines indicate the optimal value function on original MDP.



(a) Q (b) cumulative reward

Fig. 7: (a) Convergence of value function by CenQ-learning for $n = 4$ state chain walk and (b) cumulative rewards obtained by CenQ and Q-learning for $n = 10$ state chain walk. Average of 10 runs is shown.

REFERENCES

[1] Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. 2005.

[2] Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*. 1998.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[4] Minne Li, Zhiwei Qin, Yan Jiao, Yaodong Yang, Jun Wang, Chenxi Wang, Guobin Wu, and Jieping Ye. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In *The World Wide Web Conference*, pages 983–994, 2019.

[5] Sandeep Chinchali, Pan Hu, Tianshu Chu, Manu Sharma, Manu Bansal, Rakesh Misra, Marco Pavone, and Sachin Katti. Cellular network traffic scheduling with deep reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2018.

[6] Wade Genders and Saiedeh Razavi. Using a deep reinforcement learning agent for traffic signal control. *arXiv preprint arXiv:1611.01142*, 2016.

[7] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *International Conference on Knowledge Discovery & Data Mining*, pages 2496–2505, 2018.

[8] Shumpei Kubosawa, Takashi Onishi, and Yoshimasa Tsuruoka. Synthesizing chemical plant operation procedures using knowledge, dynamic simulation and deep reinforcement learning. In *SICE Annual Conference*, pages 1376–1379, 2018.

[9] Matthias Heger. Consideration of risk in reinforcement learning. In *International Conference on Machine Learning*, pages 105–111. 1994.

[10] Makoto Sato and Shigenobu Kobayashi. Variance-penalized reinforcement learning for risk-averse asset allocation. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 244–249, 2000.

[11] Oliver Mihatsch and Ralph Neuneier. Risk-sensitive reinforcement learning. *Machine learning*, 49(2-3):267–290, 2002.

[12] Peter Geibel and Fritz Wysotzki. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, 24:81–108, 2005.

[13] Javier Garcia and Fernando Fernández. Safe exploration of state and action spaces in reinforcement learning. *Journal of Artificial Intelligence Research*, 45:515–564, 2012.

[14] William Saunders, Girish Sastry, Andreas Stuhlmueller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. In *International Conference on Autonomous Agents and MultiAgent Systems*, pages 2067–2069, 2018.

[15] Theodore J Perkins and Andrew G Barto. Lyapunov design for safe reinforcement learning. *Journal of Machine Learning Research*, 3:803–832, 2002.

[16] Anayo K Akametalu, Jaime F Fisac, Jeremy H Gillula, Shahab Kaynama, Melanie N Zeilinger, and Claire J Tomlin. Reachability-based safe learning with gaussian processes. In *53rd IEEE Conference on Decision and Control*, pages 1424–1431, 2014.

[17] Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

[18] Paul Lévy. Systèmes markoviens et stationnaires. cas dénombrable. *Ann. Sci. École Norm. Sup.*, 68(3):327–381, 1951.

[19] Paul Lévy. Complément à l'étude des processus de markoff. *Ann. Sci. École Norm. Sup.*, 69(3):203–212, 1952.

[20] Paul Lévy. Processus markoviens et stationnaires. cas dénombrable. *Ann. Inst. H. Poincaré*, 18:7–25, 1958.

[21] Steven J Bradtke and Michael O Duff. Reinforcement learning methods for continuous-time markov decision problems. In *Advances in neural information processing systems*, pages 393–400, 1995.

[22] Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[23] Richard S Sutton. TD models: Modeling the world at a mixture of time scales. In *International Conference on Machine Learning*, pages 531–539. Elsevier, 1995.

[24] Doina Precup and Richard S Sutton. Multi-time models for temporally abstract planning. In *Advances in neural information processing systems*, pages 1050–1056, 1998.

[25] Ronald Edward Parr. *Hierarchical control and learning for Markov decision processes*. PhD thesis, University of California, Berkeley, 1998.

[26] David Freedman. *Approximating countable Markov chains*. Springer-Verlag New York, 1983.

[27] Y. Quennel Zhao and Danielle Liu. The censored markov chain and the best augmentation. *Journal of Applied Probability*, 33(3):623–629, 1996.

[28] John G Kemeny, J Laurie Snell, and Anthony W Knapp. *Denumerable Markov chains*. Springer-Verlag New York, 1976.