

## **AlwaysSafe: Reinforcement Learning without Safety Constraint Violations during Training**

Simão, T. D.; Jansen, Nils; Spaan, M.T.J.

**Publication date**

2021

**Document Version**

Final published version

**Published in**

Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021, IFAAMAS

**Citation (APA)**

Simão, T. D., Jansen, N., & Spaan, M. T. J. (Accepted/In press). AlwaysSafe: Reinforcement Learning without Safety Constraint Violations during Training. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021, IFAAMAS*

**Important note**

To cite this publication, please use the final published version (if applicable). Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

# AlwaysSafe: Reinforcement Learning without Safety Constraint Violations during Training – Supplementary Material

Thiago D. Simão<sup>1</sup>, Nils Jansen<sup>2</sup>, Matthijs T. J. Spaan<sup>3</sup>

<sup>1</sup> Delft University of Technology, The Netherlands  
t.diassimao@tudelft.nl

<sup>2</sup> Radboud University, Nijmegen  
n.jansen@science.ru.nl

<sup>3</sup> Delft University of Technology, The Netherlands  
m.t.j.spaan@tudelft.nl

This document provides the supplementary material of the following paper:

Thiago D. Simão, Nils Jansen, and Matthijs T. J. Spaan. 2021. AlwaysSafe: Reinforcement Learning without Safety Constraint Violations during Training. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 10 pages.

## S-I PROOF OF THEOREM 3.2

We restate the theorem for clarity.

**THEOREM (3.2).**  $\phi_C$  is cost-model-irrelevant.

**PROOF.** Given  $a, s_1, s_2, \bar{s} \in \mathbb{A} \times \mathbb{S} \times \mathbb{S} \times \bar{\mathbb{S}}$ . If  $\phi_C(s_1) = \phi_C(s_2)$  then we have

$$\begin{aligned} C(s_1, a) &= \sum_{i \in \mathbb{N}_n} C_i(s_1[\Delta_i^C], a) \\ &= \sum_{i \in \mathbb{N}_n} C_i(s_2[\Delta_i^C], a) \\ &= C(s_2, a). \end{aligned}$$

The first and last derivations come from the definition of the factored cost function. The middle derivation comes from the fact that both states were mapped together, so from (4) we conclude that  $s_1[\Delta_i^C] = s_2[\Delta_i^C] : \forall i \in \mathbb{N}_n$ .

In the following derivation, we use  $P(s'[\Delta] \mid s, a) = \prod_{X_i \in \Delta} P(s'[X_i] \mid s, a)$  where  $\Delta \subseteq X, s, a, s' \in \mathbb{S} \times \mathbb{S} \times \mathbb{A}$  and  $\text{NotAnc}(C) = X \setminus \text{Anc}(C)$ .

If  $\phi_C(s_1) = \phi_C(s_2)$  then we have

$$\begin{aligned} \sum_{s' \in \phi^{-1}(\bar{s})} P(s' \mid s_1, a) &\stackrel{(a)}{=} \sum_{s' \in \phi^{-1}(\bar{s})} P(s'[X] \mid s_1, a), \\ &\stackrel{(b)}{=} \sum_{s' \in \phi^{-1}(\bar{s})} P(s'[\text{Anc}(C)] \mid s_1, a) P(s'[\text{NotAnc}(C)] \mid s_1, a), \\ &\stackrel{(c)}{=} \sum_{s' \in \phi^{-1}(\bar{s})} P(\bar{s}[\text{Anc}(C)] \mid s_1, a) P(s'[\text{NotAnc}(C)] \mid s_1, a), \\ &\stackrel{(d)}{=} P(\bar{s}[\text{Anc}(C)] \mid s_1, a) \underbrace{\sum_{s' \in \phi^{-1}(\bar{s})} P(s'[\text{NotAnc}(C)] \mid s_1, a)}_{=1 \text{ sum over all values of NotAnc}(C)}, \\ &\stackrel{(e)}{=} P(\bar{s}[\text{Anc}(C)] \mid s_1, a) \\ &\stackrel{(f)}{=} \prod_{X_i \in \text{Anc}(C)} P(\bar{s}[X_i] \mid s_1[\text{Pa}_a(X_i)], a) \\ &\stackrel{(g)}{=} \prod_{X_i \in \text{Anc}(C)} P(\bar{s}[X_i] \mid s_2[\text{Pa}_a(X_i)], a) \\ &\stackrel{(h)}{=} \sum_{s' \in \phi^{-1}(\bar{s})} P(s' \mid s_2, a). \end{aligned}$$

In this derivation,

(a) shows we are considering the values of each variable in state  $s'$ ;

- (b) decouples ancestors from non ancestors;
- (c) removes the dependence on the state  $s'$ , since  $\forall s' \in \phi_C^{-1}(\bar{s})$ , the values of variables in  $\text{Anc}(C)$  are the same, by the definition of  $\phi_C$ ;
- (d) factors out the probability term, since it is now independent of  $s'$ ;
- (e) removes the summation that results in 1;
- (f) uses the conditional independence from the factored CMDP;
- (g) swaps  $s_1$  and  $s_2$ , since they were mapped to the same abstract state, the values of their parents are the same;
- (h) uses the same reasoning from (f) to (a).

□

## S-II PROOF OF THEOREM 4.4

We restate the theorem for clarity.

**THEOREM (4.4).** *Given an abstract CMDP built according to a cost-model irrelevance abstraction and a fixed  $\delta \in (0, 1)$ , the algorithm AlwaysSafe equipped with policies  $\pi_A$ ,  $\pi_T$  or  $\pi_\alpha$  has no constraint violation regret with probability  $1 - \delta$ .*

**PROOF.** We split the proof in three parts related to the three policies considered.

**AlwaysSafe  $\pi_A$**  From Theorem 4.1 we know that

$$V_C^{\pi_A}(\mu) \leq \hat{c}.$$

This is enough to conclude that AlwaysSafe with  $\pi_A$  does not violate the safety constraints.

**AlwaysSafe  $\pi_T$**  First let us define the maximum expected cost of executing the policy  $\pi_G$  in an CMDP of the uncertainty set:

$$\text{maxC} = \max_{P' \in \Xi} V_C^{\pi_G}(\mu, P').$$

From (7) we must show that  $\pi_T$  is safe in both cases.

- Case 1 ( $\text{maxC} \leq \hat{c}$ ): in this case the policy executed is  $\pi_G$ . This way, we have that the expected cost for executing  $\pi_G$  in any of the CMDP of the uncertainty set is smaller than  $\text{maxC}$ :

$$V_C^{\pi_G}(\mu, P') \leq \text{maxC} : \forall P' \in \Xi.$$

Therefore, if the true CMDP is in the uncertainty set, then the expected cost of the policy  $\pi_G$  is less or equal to the cost bound:

$$P \in \Xi \implies V_C^{\pi_G}(\mu) \leq \text{maxC} \leq \hat{c}, \quad (8)$$

where the last inequality comes from the condition of this case. By construction, the transition function of the true CMDP belongs to the uncertainty set  $\Xi$  with high probability  $1 - \delta$ :

$$\Pr(P \in \Xi) \geq 1 - \delta. \quad (9)$$

Therefore, we have from (8) and (9) that:

$$\begin{aligned} \Pr(V_C^{\pi_G}(\mu) \leq \hat{c}) &\geq \Pr(P \in \Xi) \\ &\geq 1 - \delta. \end{aligned}$$

- Case 2 ( $\text{maxC} > \hat{c}$ ): in this case the policy executed is  $\pi_A$ , which is safe (Theorem 4.1).

**AlwaysSafe  $\pi_\alpha$**  This proof is similar to the proof for AlwaysSafe  $\pi_T$ . In this case, the only difference is that  $\pi_G$  might be computed with a bound  $\beta\hat{c}$  that is lower than the original bound  $\hat{c}$ .

□

### S-III ENVIRONMENTS FROM THE EMPIRICAL ANALYSIS

#### S-III.1 Cliff environment

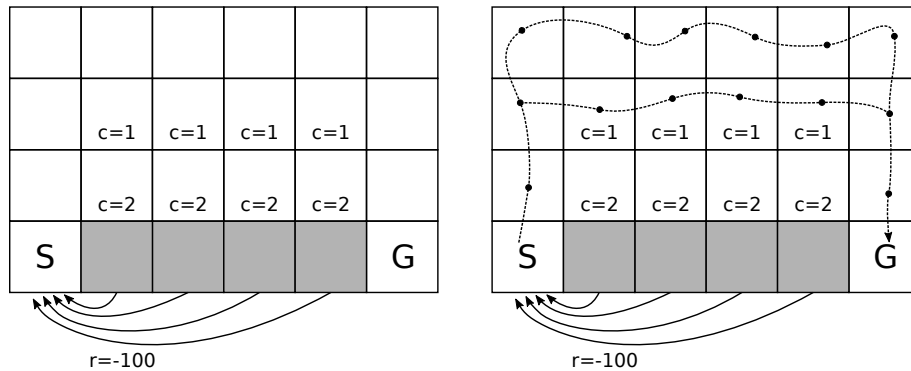


Figure 4: Cliff world.

In the cliff environment (Figure 4 left) the agent starts in position  $S$  and must reach position  $G$  (an absorbing state with no cost or reward). The agent also gets a reward of  $-1$  for each movement. If the agent falls from the cliff (stepping in one of the grey areas), it is sent back to state  $S$  and gets a reward of  $-100$ .

The agent gets a cost of  $2$  for walking in cells adjacent to the cliff (second row) and a cost of  $1$  for walking  $2$  cells away from the cliff (third row).

Figure 4 (right) shows the optimal paths (dashed lines) for a cost bound  $\hat{c} = 2$ . The agent needs to randomize between two paths, taking each path  $50\%$  of the time, which gives an expected cost of  $2$  and expected value of  $10$ .

#### S-III.2 Simple CMDP

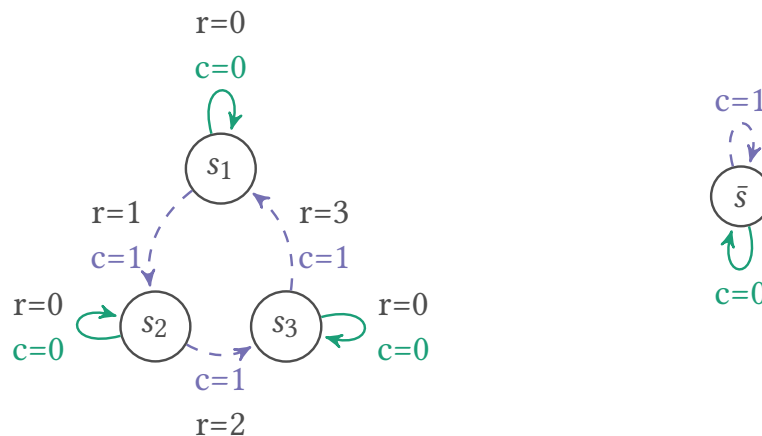


Figure 5: A CMDP with 3 states (left) and the corresponding abstract CMDP built with a model-cost-irrelevant abstraction (right).

The simple CMDP (Figure 5) was adapted from a problem proposed by Zheng and Ratliff [55]. It has 3 states and 2 actions. The agent can move from one state to the other, which give a cost of  $1$  and a reward equals to the index of the current state. Therefore, the agent has to balance between the actions move and stay to get the maximum reward without violating the cost constraints. Finally, since the reward for moving changes from one state to the other, the optimal policy is not equal in all the ground states.

## S-IV CONFIDENCE INTERVALS FROM EXPERIMENTS

We use the following confidence intervals in the experiments:

$$e^P(s, a, s') = \frac{1}{\max\{n(s, a), 1\}} + \sqrt{\frac{\text{Var}(\hat{P}(s' | s, a))}{\max\{n(s, a), 1\}}} \text{ and}$$
$$e^R(s, a) = \frac{R_{\max} - R_{\min}}{\max\{n(s, a), 1\}},$$
$$e^C(s, a) = \frac{C_{\max} - C_{\min}}{\max\{n(s, a), 1\}},$$

where

- $n(s, a)$  is the number of times action  $a \in \mathbb{A}$  has been executed in the state  $s \in \mathbb{S}$ ,
- $R_{\min}$  and  $R_{\max}$  ( $C_{\min}$  and  $C_{\max}$ ) are the minimum and maximum value of the reward (cost) function,
- $\text{Var}(x) = x * (1 - x)$ .

We removed the subscript  $\delta$  since these bounds are tighter than the theoretical bounds and do not depend on  $\delta$ .

## S-V CODE

The code to reproduce the experiments is available at <https://github.com/AlgTUDelft/AlwaysSafe>. The interested reader can follow the instructions in the README.md file to install and run the scripts for each experiment.