# Safe Learning in Tree-Form Sequential Decision Making: Handling Hard and Soft Constraints

**Martino Bernasconi** [1] **Federico Cacciamani** [1] **Matteo Castiglioni** [1] **Alberto Marchesi** [1] **Nicola Gatti** [1] **Francesco Trovò** [1]

## Abstract

We study *decision making problems* in which an agent *sequentially* interacts with a *stochastic* environment defined by means of a *tree structure*. The agent repeatedly faces the environment over time, and, after each round, it perceives a *utility* and a *cost*, which are both stochastic. The goal of the agent is to learn an optimal strategy in an online fashion, while keeping costs below a given *safety* threshold at the same time. Our model naturally fits many real-world scenarios, such as, *e.g.*, opponent exploitation in games and web link selection. We study the *hard-threshold* problem of achieving sublinear regret while guaranteeing that the threshold constraint is satisfied at every iteration with high probability. First, we show that, in general, any algorithm with such a guarantee incurs in a linear regret. This motivates the introduction of a relaxed problem, called the *soft-threshold* problem, in which we only require that the cumulative violation of the threshold constraint grows sublinearly, and, thus, we can provide an algorithm with sublinear regret. Next, in the hard-threshold problem, we show how a sublinear regret algorithm can be designed under the additional assumption that there exists a known strategy strictly satisfying the threshold constraint. We also show that our regret bounds are tight. Finally, we cast the opponent exploitation problem to our model, and we experimentally evaluate our algorithms on a standard testbed of sequential games.

## 1. Introduction

*Tree-form sequential decision making* models situations in which an agent interacts with an environment in a multi-stage process. The agent-environment interaction is sequential, and it can be represented with a tree structure made by: (i) decision nodes where the agent takes actions; and (ii) observation nodes in which the agent observes signals from the environment. This kind of interaction naturally models the decision-making problem faced by an agent in extensive-form (*i.e.*, sequential) games, which have recently received a terrific attention from the AI and machine learning community; see, *e.g.*, the superhuman agents developed for two-player (Brown & Sandholm, 2018) and multi-player (Brown & Sandholm, 2019) no-limit Texas hold'em Poker, Go (Silver et al., 2016), and Starcraft II (Vinyals et al., 2019). Moreover, it also fits many real-world scenarios, such as, *e.g.*, multi-level web link selection, path routing in telecommunication networks, robot patrolling, and mission planning in military settings.

In spite of the popularity gained by sequential decision making problems over the last years, most of the works on the topic focus on the basic (unconstrained) online learning problem faced by a utility-maximizing agent that repeatedly plays the decision making process. However, these works do *not* account for the case, which is common in safety-critical systems, where agent's decisions at each round are subject to *safety constraints* depending on unknown parameters. For instance, these constraints are crucial for many cyber-physical systems with humans in the loop, such as, *e.g.*, autonomous driving, power grids management, and opponent exploitation in games. While safety constraints have been widely studied in one-shot decision making problems (see, *e.g.*, (Usmanova et al., 2019)), to the best of our knowledge our work is the first one addressing online learning with safety constraints in general tree-form sequential decision making settings.

### 1.1. Original Contributions

We study *stochastic tree-form sequential decision making processes with costs* (compactly referred to as SDMCs). In an SDMC, at the end of each sequential interaction, the agent perceives a utility and a cost. In our model, both utilities and costs are stochastic, as well as how the environment evolves determining signals perceived by the agent

[1]Politecnico di Milano, Milan, Italy. Correspondence to: Martino Bernasconi <martino.bernasconideluca@polimi.it>.

*Proceedings of the $39^{th}$ International Conference on Machine Learning*, Baltimore, Maryland, USA, PMLR 162, 2022. 

at observation nodes. The goal of the agent is to learn an optimal strategy in an online fashion by repeatedly playing the SDMC, while keeping the expected cost below a given *safety threshold* at the same time. In this work, we make minimal assumptions on the feedback received by the agent, which, beside realized utilities and costs, only encompasses the sequence of decision nodes and signals encountered while playing the SDMC.

We address the *hard-threshold* problem of achieving sublinear regret in the number of rounds $T$ while being $\delta$-safe for any given $\delta \in (0, 1)$, which means that the threshold constraint is satisfied at every iteration with probability at least $1 - \delta$. We provide an impossibility result for the hard-threshold problem stating that any $\delta$-safe algorithm must incur in a linear regret with high probability. This motivates the introduction of a relaxed problem, called the *soft-threshold* problem, in which we only require that the cumulative violation of the threshold constraint grows sublinearly. For this problem, we provide a UCB-inspired algorithm attaining sublinear regret, which, at each round, plays the SDMC according to a randomized strategy obtained by solving a *linear program* (LP) that computes an optimal and safe agent's strategy using upper and lower bounds for utilities and costs, respectively. Such bounds are built using specifically crafted estimates of utilities and costs, so as to leverage the tree structure of the SDMC and formulate the optimization problem as an LP. Then, we switch back to the hard-threshold problem, where we show that, surprisingly, our initial negative result can be circumvented by introducing the additional, reasonable assumption that the agent knows a strategy that is always *strictly* safe. Such an assumption allows us to design a $\delta$-safe algorithm with sublinear regret, which works by solving the same LP used in the soft-threshold algorithm, but plays a different strategy that is obtained by properly combining the solution of the LP with the always-strictly-safe strategy. Moreover, for both our algorithms, we provide lower bounds showing that their guarantees are tight. Finally, we cast the *utility-constrained opponent exploitation* problem recently introduced by Bernasconi-de-Luca et al. (2021) to our model, and we experimentally evaluate our algorithms on a standard testbed of sequential games.

### 1.2. Related Works

The problem of safe learning has been widely studied in the online learning literature, even if most of the works on the topic consider one-shot decision making problems. See, *e.g.*, the works on multi-armed bandits with linear constraints (Chen et al., 2018; Saxena et al., 2020) and their extension to linear bandits (Usmanova et al., 2019; Amani et al., 2020; Liu et al., 2021; Pacchiano et al., 2021). To the best of our knowledge, the only work addressing safe learning in a multi-armed bandit with a sequential structure is that by Chen et al. (2018). However, they consider a specific structure, which is a bi-level decision making problem modeling a web link selection process. Our model subsumes theirs and it can be applied to much more general sequential settings. Let us also remark that the techniques developed for safe learning in linear bandits assume a specific structure of the feedback, which only depends on the agent's decision, a fixed environment parameter, and an additional random noise. Instead, our model assumes that each feedback is determined by the signals sampled by the environment during the sequential decision process. Even if these two feedbacks are equivalent in expectation, they are different random variables, and this renders the techniques used for linear bandits inapplicable in our setting.

Tree-form sequential decision making problems were originally introduced by Farina et al. (2021) and Farina & Sandholm (2021), who focus on adversarial decision-making problems. Our setting significantly differs from theirs in two crucial aspects: (i) we add costs, which are associated to terminal nodes of the tree structure; and (ii) we assume that the environment is stochastic, *i.e.*, that utilities, costs, and signals are randomly drawn according to some probability distributions. Notice that assuming an adversarial environment in our setting with costs makes our goal of designing $\delta$-safe algorithms unfeasible, unless one resorts to strong, unreasonable assumptions. Indeed, the features of SMDCs render our learning problem considerably different from the adversarial problems described above, and, thus, our techniques are more akin to those used in safe online learning than those adopted by Farina et al. (2021) and Farina & Sandholm (2021).

## 2. Preliminaries

In SDMCs, a utility-maximizing agent interacts with a stochastic environment by taking sequential decisions subject to costs. The interaction underlying an SDMC is defined by a finite tree, whose set of nodes $\mathcal{K}$ is partitioned into three disjoint subsets: (i) the set $\mathcal{I}$ of *decision nodes*, in which the agent takes decisions; (ii) the set $\mathcal{J}$ of *observation nodes*, where the agent receives signals from the environment; and (iii) the set $\mathcal{Z}$ of *terminal nodes*, in which the SDMC ends. For every decision node $i \in \mathcal{I}$, we denote by $A_i$ the finite set of *actions* available to the agent at $i$, while, for every observation node $j \in \mathcal{J}$, we let $S_j$ be the finite set of possible *signals* that the agent may receive at $j$.

The dynamics of an SDMC is as follows. When the agent takes an action in a decision node $i \in \mathcal{I}$, then the process transitions to one of the children of node $i$, depending on the chosen action $a \in A_i$. The same happens, on the environment side, for a given observation node $j \in \mathcal{J}$ and signal $s \in S_j$. The interaction ends whenever a terminal node $z \in \mathcal{Z}$ is reached. The environment draws signals at each

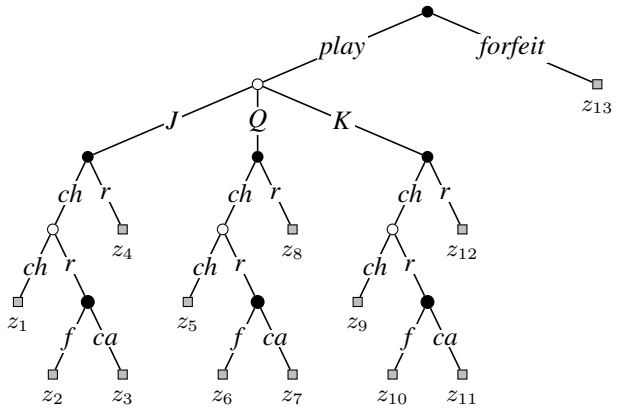


*Figure 1.* Example of an SDMC representing a Kuhn Poker game with 3 ranks and an additional action (*forfeit*). Black rounded nodes represent decision nodes, white rounded nodes are observation nodes, and gray squared nodes are terminal nodes. Labels on the edges correspond to actions played by the agent or signals sent by the environment (*ch* stands for *check*, *r* for *raise*, *f* for *fold*, and *ca* for *call*). Finally, *J*, *Q*, and *K* represent the card dealt by the environment to the agent, *i.e.*, *Jack*, *Queen*, and *King*, respectively.

observation node according to some probability distribution that is unknown to the agent. Formally, we let $\rho_{js} \in [0,1]$ be the probability of signal $s \in S_j$ at observation node $j \in \mathcal{J}$, so that $\sum_{s \in S_j} \rho_{js} = 1$ for all $j \in \mathcal{J}$. In what follows, by a slight abuse the notation, given any terminal node $z \in \mathcal{Z}$ we let $\rho(z)$ be the probability of reaching $z$ due to environment transitions only, *i.e.*, $\rho(z)$ is the product of the probabilities $\rho_{js}$ of all the pairs $j \in \mathcal{J}$, $s \in S_j$ encountered on the path from the root of the tree to $z$. Figure 1 shows an example of SMDC corresponding to the tree of Kuhn poker with 3 cards (Kuhn, 2016). In an SDMC, utilities and costs are unknown to the agent and modeled by means of bounded random variables. Formally, each terminal node $z \in \mathcal{Z}$ is associated with two random variables, namely $U_z$ and $C_z$, which are both bounded in the interval $[a, b]$. For ease of notation, we let $\Delta \coloneqq b - a$.

Any node $k \in \mathcal{K}$ identifies a *sequence* of agent's actions, which are those encountered on the path from the root of the tree to that node. We denote such a sequence as $\sigma(k)$. As customary in the literature (see, *e.g.*, (Farina et al., 2021)), we adopt the notation $(i, a)$ to denote the sequence obtained by appending action $a \in A_i$ to the sequence $\sigma(i)$ identified by node $i \in \mathcal{I}$. Then, $\Sigma \coloneqq \{(i,a) \,|\, i \in \mathcal{I},\, a \in \mathcal{A}_i\} \cup \{\varnothing\}$ is the set of all sequences, where $\varnothing$ is the *empty sequence* identified by the root of the tree. With a slight abuse of notation, we use $\sigma$ to denote a generic sequence $\sigma \in \Sigma$.

**Strategy Representation.** An agent's *strategy* in an SDMC defines a probability distribution over the actions that are available at each decision node. In this work, we represent strategies by exploiting the widely-used *sequence-form representation* (von Stengel, 1996; Koller et al., 1996). In such a representation, a strategy is encoded by a vector $\boldsymbol{x} \in \mathbb{R}^{|\Sigma|}_{\geq 0}$ indexed over the set of sequences $\Sigma$, where, intuitively, the element $\boldsymbol{x}[\sigma]$ associated with sequence $\sigma \in \Sigma$ expresses the realization probability of that specific sequence of agent's actions. Then, a vector $\boldsymbol{x} \in \mathbb{R}^{|\Sigma|}_{\geq 0}$ is a valid sequence-form strategy if it satisfies the following linear constraints:

$$\boldsymbol{x}[\varnothing] = 1, \quad \boldsymbol{x}[\sigma(i)] = \sum_{a \in \mathcal{A}_i} \boldsymbol{x}[(i,a)] \quad \forall i \in \mathcal{I}. \tag{1}$$

The set of linear constraints in Equation (1) characterizes the polytope of all the valid sequence-form strategies, which, from now on, is denoted as $\mathcal{X}$. For compactness, we introduce a matrix $\boldsymbol{F} \in \{-1, 0, 1\}^{|\mathcal{I}| \times |\Sigma|}$ and a vector $\boldsymbol{f} \in \{0,1\}^{|\mathcal{I}|}$ such that the sequence-form polytope is formally defined as:[1]

$$\mathcal{X} \coloneqq \left\{ \boldsymbol{x} \in \mathbb{R}^{|\Sigma|}_{\geq 0} \mid \boldsymbol{F}\boldsymbol{x} = \boldsymbol{f} \right\}.$$

Furthermore, let $\Pi \coloneqq \mathcal{X} \cap \{0,1\}^{|\Sigma|}$ be the set of agent's *pure strategies*, *i.e.*, those deterministically prescribing a single action at each decision node. It is immediate to check that $\mathcal{X}$ can be written as the convex hull of the set of pure strategies, namely $\mathcal{X} \coloneqq \operatorname{co} \Pi$.

**Expected Utilities and Costs.** The sequence-form representation allows us to write the expected utilities and costs as *linear* functions of the agent's strategy. Given $\boldsymbol{x} \in \mathcal{X}$, the expected utility $u(\boldsymbol{x})$, respectively the expected cost $c(\boldsymbol{x})$, is the sum of the expectations of random variables $U_z$, respectively $C_z$, over all the terminal nodes $z \in \mathcal{Z}$, weighted by the probability of reaching such terminal nodes. Formally:

$$u(\boldsymbol{x}) \coloneqq \sum_{z \in \mathcal{Z}} \boldsymbol{x}[\sigma(z)]\, \rho(z)\, \mathbb{E}[U_z],$$
$$c(\boldsymbol{x}) \coloneqq \sum_{z \in \mathcal{Z}} \boldsymbol{x}[\sigma(z)]\, \rho(z)\, \mathbb{E}[C_z].$$

The expected utility $u(\boldsymbol{x})$ can be re-written as follows:

$$u(\boldsymbol{x}) = \sum_{\sigma \in \Sigma} \boldsymbol{x}[\sigma] \underbrace{\left( \sum_{\substack{z \in \mathcal{Z}: \\ \sigma(z) = \sigma}} \rho(z)\, \mathbb{E}[U_z] \right)}_{=: \boldsymbol{\theta}^\star[\sigma]} = \boldsymbol{x}^\top \boldsymbol{\theta}^\star,$$

where the vector $\boldsymbol{\theta}^\star \in \mathbb{R}^{|\Sigma|}$ includes all the quantities related to the environment appearing in the definition of $u(\boldsymbol{x})$. Similarly, we write $c(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\omega}^\star$ by introducing the vector $\boldsymbol{\omega}^\star \in \mathbb{R}^{|\Sigma|}$ having elements $\boldsymbol{\omega}^\star[\sigma] \coloneqq \sum_{z \in \mathcal{Z}: \sigma(z) = \sigma} \rho(z)\, \mathbb{E}[C_z]$ for all $\sigma \in \Sigma$. Finally, let us remark that the vectors $\boldsymbol{\theta}^\star$ and $\boldsymbol{\omega}^\star$ are defined by quantities that are unknown to the agent.

[1] The formal definitions of $\boldsymbol{F}$ and $\boldsymbol{f}$ are in (von Stengel, 1996).

## 3. Problem Formulation

We study an online learning problem in which the agent repeatedly interacts with the environment in an SDMC (see Figure 2 for a graphical representation of the interaction). We denote by $T \in \mathbb{N}_{>0}$ the number of rounds of the interaction. At each round $t \in [T]$, based on the information collected up to round $t-1$, the agent selects a strategy $\boldsymbol{x}_t \in \mathcal{X}$ to be adopted at $t$.[2] Then, the sequential decision-making process is unfolded, with the agent employing a pure strategy $\boldsymbol{\pi}_t \in \Pi$ drawn according to $\boldsymbol{x}_t$.[3] The process ends upon reaching some terminal node $z_t \in \mathcal{Z}$, which is determined by $\boldsymbol{\pi}_t$ and the signals drawn by the environment at observation nodes encountered down the tree. Finally, the environment draws a utility value $u_t \sim U_{z_t}$ and a cost value $c_t \sim C_{z_t}$ from the random variables corresponding to $z_t$, and these are revealed to the agent together with $z_t$.

In a repeated SDMC, the goal of the agent is twofold. First, strategies $\boldsymbol{x}_t$ have to be selected so that, with high probability, at every round $t \in [T]$ the strategy $\boldsymbol{x}_t$ satisfies a *safety constraint* on its expected cost. In particular, we require that the expected cost $\boldsymbol{x}_t^\top \boldsymbol{\omega}^\star$ is below a given threshold $\gamma \in \mathbb{R}$. Second, the agent has to select strategies that maximize its performance (in terms of cumulative expected utility) with respect to an optimal strategy satisfying the safety constraint, *i.e.*, computed knowing the vectors $\boldsymbol{\theta}^\star$ and $\boldsymbol{\omega}^\star$.

In this work, we refer to the problem described above as the *hard-threshold* problem. Formally, we let

$$\mathcal{X}^\star \coloneqq \left\{\boldsymbol{x} \in \mathcal{X} \mid \boldsymbol{x}^\top \boldsymbol{\omega}^\star \leq \gamma\right\}$$

be the set of strategies that satisfy the safety constraint. We assume the existence of a known feasible strategy.

**Assumption 1.** *There exists a known* always-safe *strategy* $\boldsymbol{x}^\diamond \in \mathcal{X}$ *such that* $\boldsymbol{x}^{\diamond,\top} \boldsymbol{\omega}^\star \leq \gamma$.

This implies that the set $\mathcal{X}^\star$ is always non-empty. Then, we measure the performance of the agent after $T$ rounds by means of the following notion of regret:

$$R_T \coloneqq \max_{\boldsymbol{x} \in \mathcal{X}^\star} \sum_{t=1}^{T} \boldsymbol{x}^\top \boldsymbol{\theta}^\star - \sum_{t=1}^{T} \boldsymbol{x}_t^\top \boldsymbol{\theta}^\star.$$

Moreover, we let $v_t \coloneqq \boldsymbol{x}_t^\top \boldsymbol{\omega}^\star - \gamma$ be the *violation* of the safety constraint during round $t \in [T]$. Thus, we formalize the agent's goal as that of designing an online learning algorithm such that: (i) its regret grows sublinearly in the number of rounds $T$, *i.e.*, $R_T = o(T)$; and (ii) it is $\delta$-*safe* for any $\delta \in (0,1)$ given as input, *i.e.*, with probability at least $1-\delta$, it holds that $v_t \leq 0$ for all $t \in [T]$.

As we show next, no algorithm can achieve both (i) and (ii) without requiring additional assumptions on the SDMC setting. Thus, we also introduce a relaxed version of the hard-threshold problem, which we call *soft-threshold* problem. In particular, we relax point (ii) by requiring instead that the sum of the positive violations $v_t$ incurred by the algorithm grows sublinearly in the number of rounds. Formally, we define the *cumulative violation* up to round $T$ as follows:

$$V_T \coloneqq \sum_{t=1}^{T} (v_t)^+,$$

where $(v_t)^+ \coloneqq \max\{v_t, 0\}$, so that we can express the requirement of the soft-threshold problem as $V_T = o(T)$.

## 4. Impossibility Result

We start by providing a negative result for the hard-threshold problem. In particular, the following theorem shows that, in general repeated SDMCs, no learning algorithm for the agent can achieve sublinear regret in the number of rounds $T$ while, at the same time, being $\delta$-safe for any $\delta \in (0,1)$ given as input. This motivates our study of the soft-threshold problem in Section 6. Nevertheless, in Section 7 we show that the negative result stated in the following theorem can be circumvented by introducing an additional assumption on the SDMC.[4]

**Theorem 1.** *In general repeated SDMCs, if an algorithm is* $\delta$*-safe for any* $\delta \in (0,1)$ *given as input, then it incurs in a regret* $R_T = \Omega(T)$ *with probability at least* $1-\delta$.

## 5. Parameters Estimation

The algorithms that we propose in the following sections rely on having access to unbiased estimators and high-probability confidence bounds for the vectors $\boldsymbol{\theta}^\star$ and $\boldsymbol{\omega}^\star$. This section is devoted to introducing these tools.

For ease of notation, for every sequence $\sigma \in \Sigma$ and round $t \in [T]$, we define the following two subsets:

$$\tau_t^{ter}(\sigma) \coloneqq \{\tau \in [t-1] \mid \sigma(z_\tau) = \sigma\},$$
$$\tau_t^{pure}(\sigma) \coloneqq \{\tau \in [t-1] \mid \boldsymbol{\pi}_\tau[\sigma] = 1\}.$$

Intuitively, $\tau_t^{ter}(\sigma)$ contains all the rounds up to $t-1$ in which the actions prescribed by $\sigma$ are those actually played by the agent during the SDMC, while $\tau_t^{pure}(\sigma)$ defines all the rounds up to $t-1$ in which the pure strategy drawn by the agent prescribed to play all the actions in $\sigma$ (these are played or *not* depending on the environment transitions). Furthermore, we let $n_t(\sigma) \coloneqq |\tau_t^{pure}(\sigma)|$ for all $\sigma \in \Sigma$.

[2] For the ease of notation, we compactly denote by $[n]$ the set $\{1, \dots, n\}$ made by the first $n \in \mathbb{N}_{>0}$ natural numbers.

[3] The pure strategy $\boldsymbol{\pi}_t$ is drawn according to $\boldsymbol{x}_t$ by applying a sampling scheme that selects an action at each decision node $i \in \mathcal{I}$ of the tree according to the probability distribution defined as $\frac{\boldsymbol{x}_t[(i,a)]}{\boldsymbol{x}_t[\sigma(i)]}$ for $a \in A_i$ (see, *e.g.*, (Farina et al., 2021)).

[4] All the proofs are in the Appendix.

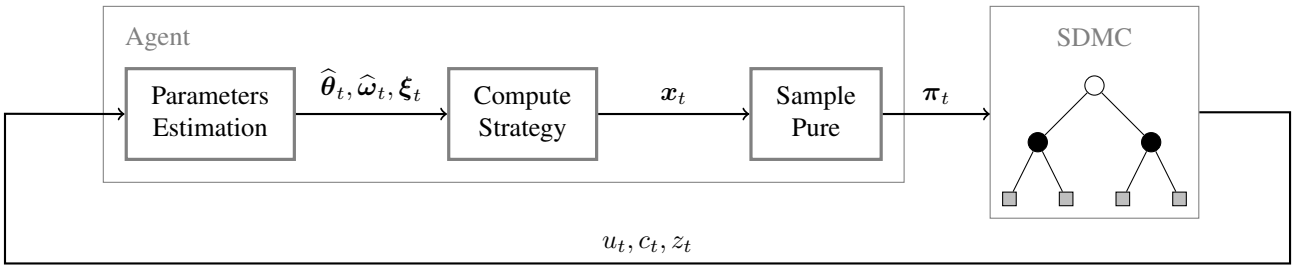


*Figure 2.* Overview of the interaction between the agent and the environment in an SDMC.

### 5.1. Unbiased Estimators

Given the agent's observations up to round $t-1$, we define the estimators $\widehat{\boldsymbol{\theta}}_t$ and $\widehat{\boldsymbol{\omega}}_t$ for the vectors $\boldsymbol{\theta}^\star$ and $\boldsymbol{\omega}^\star$, respectively, so that, for every $\sigma \in \Sigma$:

$$\widehat{\boldsymbol{\theta}}_t[\sigma] \coloneqq \frac{1}{n_t(\sigma)} \sum_{\tau \in \tau_t^{ter}(\sigma)} u_\tau, \quad \widehat{\boldsymbol{\omega}}_t[\sigma] \coloneqq \frac{1}{n_t(\sigma)} \sum_{\tau \in \tau_t^{ter}(\sigma)} c_\tau.$$

Given how $\boldsymbol{\theta}^\star$ and $\boldsymbol{\omega}^\star$ are defined, $\widehat{\boldsymbol{\theta}}_t$ and $\widehat{\boldsymbol{\omega}}_t$ need to jointly estimate the probabilities $\rho(z)$ and the expected values of the random variables defining utilities and costs, respectively. Intuitively, this is achieved by writing the component $\widehat{\boldsymbol{\theta}}_t[\sigma]$ (or $\widehat{\boldsymbol{\omega}}_t[\sigma]$), associated with sequence $\sigma \in \Sigma$ as the average of the observations of the utilities (or costs) obtained at each round $\tau \in [t-1]$ such that the pure strategy $\boldsymbol{\pi}_\tau$ prescribed actions in $\sigma$, where each observation is weighted by an indicator function that is equal to 1 if and only if the terminal node $z_\tau$ reached at round $\tau$ is such that $\sigma(z_\tau) = \sigma$ (*i.e.*, the agent actually played all the actions in $\sigma$ at $\tau$).

The observation above is crucial to prove the following lemma, which shows that $\widehat{\boldsymbol{\theta}}_t$ and $\widehat{\boldsymbol{\omega}}_t$ are unbiased estimators of $\boldsymbol{\theta}^\star$ and $\boldsymbol{\omega}^\star$, respectively.

**Lemma 1.** *For any $t \in [T]$, $\mathbb{E}\left[\widehat{\boldsymbol{\theta}}_t\right] = \boldsymbol{\theta}^\star$ and $\mathbb{E}\left[\widehat{\boldsymbol{\omega}}_t\right] = \boldsymbol{\omega}^\star$.*

### 5.2. High-Probability Confidence Bounds

By observing that the components of $\widehat{\boldsymbol{\theta}}_t$ and $\widehat{\boldsymbol{\omega}}_t$ can be seen as empirical means computed only on a subset of observations in rounds preceding $t$, we can derive high-probability confidence bounds for $\boldsymbol{\theta}^\star$ and $\boldsymbol{\omega}^\star$, as follows:

**Lemma 2.** *Given a confidence level $\delta' \in (0,1)$, for every $t \in [T]$ and $\sigma \in \Sigma$, the following bounds hold:*

$$\mathbb{P}\left\{\left|\widehat{\boldsymbol{\theta}}_t[\sigma] - \boldsymbol{\theta}^\star[\sigma]\right| \leq \boldsymbol{\xi}_t[\sigma]\right\} \geq 1-\delta',$$
$$\mathbb{P}\left\{\left|\widehat{\boldsymbol{\omega}}_t[\sigma] - \boldsymbol{\omega}^\star[\sigma]\right| \leq \boldsymbol{\xi}_t[\sigma]\right\} \geq 1-\delta',$$

*where $\boldsymbol{\xi}_t \in \mathbb{R}_{\geq 0}^{|\Sigma|}$ is a vector such that, for every $\sigma \in \Sigma$:*

$$\boldsymbol{\xi}_t[\sigma] \coloneqq \Delta \sqrt{\frac{\log(2/\delta')}{2n_t(\sigma)}}.$$

Moreover, at each round $t \in [T]$, we define upper and lower confidence bounds for the vector $\boldsymbol{\theta}^\star$ as $\overline{\boldsymbol{\theta}}_t \coloneqq \widehat{\boldsymbol{\theta}}_t + \boldsymbol{\xi}_t$ and $\underline{\boldsymbol{\theta}}_t \coloneqq \widehat{\boldsymbol{\theta}}_t - \boldsymbol{\xi}_t$, respectively. Similarly, for the vector $\boldsymbol{\omega}^\star$, we define bounds $\overline{\boldsymbol{\omega}}_t \coloneqq \widehat{\boldsymbol{\omega}}_t + \boldsymbol{\xi}_t$ and $\underline{\boldsymbol{\omega}}_t \coloneqq \widehat{\boldsymbol{\omega}}_t - \boldsymbol{\xi}_t$.[5]

## 6. Soft–Threshold Problem

In this section, we study the soft-threshold problem in general repeated SDMCs. In Section 6.1, we provide an algorithm that achieves cumulative regret and violation that grow sublinearly in the number of rounds $T$. Then, in Section 6.2, we prove a lower bound for the problem showing that the bounds of our algorithm are tight with respect to $T$.

### 6.1. The Algorithm for the Soft–Threshold Problem

Our algorithm for the soft-threshold problem (Algorithm 1) works by applying a confidence-bound-based approach that exploits the sequence-form strategy space of SDMCs. Its core idea is to select the strategy to play at each round by considering upper and lower bounds on utilities and costs, respectively. In particular, at each round $t \in [T]$ and given a threshold $\gamma \in \mathbb{R}$, our algorithm works by solving the following linear program $\text{LP}(\overline{\boldsymbol{\theta}}_t, \underline{\boldsymbol{\omega}}_t)$ parametrized by the vector of upper bounds $\overline{\boldsymbol{\theta}}_t$ and that of lower bounds $\underline{\boldsymbol{\omega}}_t$.

$$\text{LP}(\overline{\boldsymbol{\theta}}_t, \underline{\boldsymbol{\omega}}_t) \coloneqq \begin{cases} \max\limits_{\boldsymbol{x} \succcurlyeq \boldsymbol{0}} & \boldsymbol{x}^\top \overline{\boldsymbol{\theta}}_t \quad \text{s.t.} \\ & \boldsymbol{F}\boldsymbol{x} = \boldsymbol{f} \\ & \boldsymbol{x}^\top \underline{\boldsymbol{\omega}}_t \leq \gamma \end{cases} .$$

The LP above computes an agent's strategy maximizing the upper bounds on utilities defined by $\overline{\boldsymbol{\theta}}_t$ subject to a safety constraint evaluated with the lower bounds $\underline{\boldsymbol{\omega}}_t$ on costs.

Algorithm 1 takes a threshold $\gamma$, a number of rounds $T$, a confidence $\delta \in (0,1)$, and an always-safe $\boldsymbol{x}^\diamond \in \mathcal{X}^\star$.[6] Then, at each round $t \in [T]$, the algorithm solves $\text{LP}(\overline{\boldsymbol{\theta}}_t, \underline{\boldsymbol{\omega}}_t)$. If there exists an optimal solution $\boldsymbol{x}_t$ to the linear program, it

[5] In the rest of this work, for ease of notation, we omit the dependence of $\boldsymbol{\xi}_t$, $\overline{\boldsymbol{\theta}}_t$, $\underline{\boldsymbol{\theta}}_t$, $\overline{\boldsymbol{\omega}}_t$, and $\underline{\boldsymbol{\omega}}_t$ from the confidence level $\delta' \in (0,1)$, as the latter will be specified when needed.

[6] Indeed, Algorithm 1 also works without knowing an always-safe strategy $\boldsymbol{x}^\diamond$, by playing any $\boldsymbol{x} \in \mathcal{X}$ instead.

**Algorithm 1** Algorithm for the soft-threshold problem

**Require:** $\gamma \in \mathbb{R}$, $T \in \mathbb{N}_{>0}$, $\delta \in (0,1)$, $\boldsymbol{x}^\diamond \in \mathcal{X}^\star$
  $t \leftarrow 1$;
  Initialize $\overline{\boldsymbol{\theta}}_1[\sigma] \leftarrow \infty$ and $\underline{\boldsymbol{\omega}}_1[\sigma] \leftarrow -\infty$ for all $\sigma \in \Sigma$
  **while** $t \leq T$ **do**
    **if** LP$(\overline{\boldsymbol{\theta}}_t, \underline{\boldsymbol{\omega}}_t)$ is feasible **then**
      $\boldsymbol{x}_t \leftarrow$ optimal solution to LP$(\overline{\boldsymbol{\theta}}_t, \underline{\boldsymbol{\omega}}_t)$
    **else**
      $\boldsymbol{x}_t \leftarrow \boldsymbol{x}^\diamond$
    **end if**
    Play the SDMC by using strategy $\boldsymbol{x}_t$
    Observe terminal node $z_t$, utility $u_t$, and cost $c_t$
    Compute $\overline{\boldsymbol{\theta}}_{t+1}, \underline{\boldsymbol{\omega}}_{t+1}$ using confidence $\delta' = \frac{\delta}{4T|\Sigma|}$
    $t \leftarrow t+1$
  **end while**

is used when playing the SDMC. Otherwise, if LP$(\overline{\boldsymbol{\theta}}_t, \underline{\boldsymbol{\omega}}_t)$ is unfeasible, then the algorithm adopts $\boldsymbol{x}^\diamond$, which is the always-safe strategy in $\mathcal{X}$ known to the agent by assumption. Finally, the algorithm observes $z_t$, $u_t$, and $c_t$ as feedbacks from playing the SDMC, and it employs them to update the bounds $\overline{\boldsymbol{\theta}}_{t+1}$ and $\underline{\boldsymbol{\omega}}_{t+1}$, which are used by the linear program solved by the algorithm during the next round.

Notice that, during rounds $t \in [T]$ in which LP$(\overline{\boldsymbol{\theta}}_t, \underline{\boldsymbol{\omega}}_t)$ is unfeasible, playing the alway-safe strategy $\boldsymbol{x}^\diamond$ does *not* provide any guarantee on the regret the algorithm attains. Nevertheless, as we show in our proofs, the linear program is unfeasible with low probability, and, thus, this event does *not* hinder the overall regret guarantees.

The theoretical guarantees of Algorithm 1 are stated in the following Theorem 2. Let us remark that the theorem provides cumulative regret and violation bounds that hold with high probability $1-\delta$. All of our regret bounds also hold in expectation by taking $\delta \propto \frac{1}{T}$.

**Theorem 2.** *In a general repeated SDMC, Algorithm 1 with $\delta \in (0,1)$ and $T \in \mathbb{N}_{>0}$ as inputs guarantees that, with probability at least $1-\delta$, the following bounds hold:*

$$R_T \leq 4\Delta|\Sigma|\sqrt{2T\log(2/\delta)}, \;\; V_T \leq 4\Delta|\Sigma|\sqrt{2T\log(2/\delta)}.$$

The central result that enables us to prove Theorem 2 is stated in the following technical lemma:

**Lemma 3.** *The strategies $\boldsymbol{x}_t \in \mathcal{X}$ selected by Algorithm 1 with $\delta \in (0,1)$ and $T \in \mathbb{N}_{>0}$ as inputs are such that, with probability at least $1 - \frac{\delta}{2}$, it holds:*

$$\sum_{t=1}^{T} \boldsymbol{x}_t^\top \boldsymbol{\xi}_t \leq 2\Delta|\Sigma|\sqrt{2T\log(2/\delta)}.$$

Intuitively, Lemma 3 states that the uncertainty on the utilities/costs of strategies $\boldsymbol{x}_t$ played by Algorithm 1 concentrate at a rate of $\mathcal{O}(1/\sqrt{T})$.

### 6.2. Lower Bound for the Soft–Threshold Problem

Next, we prove that the regret and violation bounds of Algorithm 1 are tight. In particular, we show that any algorithm that attains cumulative violation $V_T$ that grows "too slowly" with the number of rounds $T$ would incur in a regret linear in $T$. The following theorem formalize this statement.

**Theorem 3.** *In general repeated SDMCs, if an algorithm guarantees $V_T = o(\sqrt{T})$ with probability at least $1-\delta$ for any $\delta \in (0, \frac{1}{3})$ given as input, then it incurs in a regret $R_T = \Omega(T)$ with probability at least $1-3\delta$.*

Notice that Theorem 3 shows that the bounds achieved by Algorithm 1 are tight. Indeed, it is always the case that $R_T = \Omega(\sqrt{T})$ (it can be proven by considering an instance in which all the sequences have cost strictly less than $\gamma$), and Theorem 3 proves that one cannot improve the bound $V_T = \mathcal{O}(\sqrt{T})$ without incurring in a linear regret.

## 7. Hard–Threshold Problem

In this section, we switch the attention to the hard-threshold problem. By Theorem 1, in this case we cannot design an algorithm that is $\delta$-safe for any $\delta \in (0,1)$ while attaining sublinear regret in $T$. To circumvent such a negative result, as we show next, we need to introduce the additional, stringent assumption that the always-safe strategy $\boldsymbol{x}^\diamond \in \mathcal{X}^\star$ known to the agent is *strictly safe*. Formally:

**Assumption 2.** *There exists a known* always-strictly-safe *strategy $\boldsymbol{x}^\diamond \in \mathcal{X}$, which is such that $\boldsymbol{x}^{\diamond,\top}\boldsymbol{\omega}^\star = \gamma - \lambda$ for some $\lambda \in \mathbb{R}_{>0}$ that is known to the agent.*

Notice that Assumption 2 is reasonable in many real-world settings, where $\boldsymbol{x}^\diamond$ can be thought of as a strategy representing the case in which the agent avoids playing the SDMC.

In Section 7.1, by leveraging Assumption 2, we provide our algorithm for the hard-threshold problem, while, in Section 7.2, we show that its regret bound is tight.

### 7.1. The Algorithm for the Hard–Threshold Problem

Our algorithm for the hard-threshold problem (Algorithm 2) exploits Assumption 2 to balance the amount of exploration needed to attain sublinear regret with the requirement of being $\delta$-safe. Its core idea is to select, at each $t \in [T]$, a strategy $\boldsymbol{x}_t$ that is obtained as a convex combination of the always-strictly-safe strategy $\boldsymbol{x}^\diamond$ and a strategy $\tilde{\boldsymbol{x}}_t$ obtained by solving LP$(\overline{\boldsymbol{\theta}}_t, \underline{\boldsymbol{\omega}}_t)$. This procedure allows the algorithm to ensure that, with high probability, $\boldsymbol{x}_t$ satisfies the safety constraint, since the strategy $\tilde{\boldsymbol{x}}_t$, which is the one that Algorithm 1 would have selected, does *not* guarantee that. In particular, the mixing probability $p_t$ is chosen in such a way that, even considering the upper bounds for $\boldsymbol{\omega}^\star$, the convex combination of $\boldsymbol{x}^\diamond$ and $\tilde{\boldsymbol{x}}_t$ satisfies the safety constraint.

**Algorithm 2** Algorithm for the hard-threshold problem

**Require:** $\gamma \in \mathbb{R}, T \in \mathbb{N}_{>0}, \delta \in (0,1), \lambda > 0, \boldsymbol{x}^\diamond \in \mathcal{X}^\star$
  $t \leftarrow 1$;
  Initialize $\overline{\boldsymbol{\theta}}_1[\sigma] \leftarrow \infty, \underline{\boldsymbol{\omega}}_1[\sigma], \overline{\boldsymbol{\omega}}_1[\sigma] \leftarrow -\infty$ for $\sigma \in \Sigma$
  **while** $t \leq T$ **do**
    **if** $\mathrm{LP}(\overline{\boldsymbol{\theta}}_t, \underline{\boldsymbol{\omega}}_t)$ is feasible **then**
      $\tilde{\boldsymbol{x}}_t \leftarrow$ optimal solution to $\mathrm{LP}(\overline{\boldsymbol{\theta}}_t, \underline{\boldsymbol{\omega}}_t)$
    **else**
      $\tilde{\boldsymbol{x}}_t \leftarrow \boldsymbol{x}^\diamond$
    **end if**
    $p_t \leftarrow \begin{cases} \frac{\min\{\tilde{\boldsymbol{x}}_t^\top \overline{\boldsymbol{\omega}}_t, \Delta\} - \gamma}{\min\{\tilde{\boldsymbol{x}}_t^\top \overline{\boldsymbol{\omega}}_t, \Delta\} - \gamma + \lambda} & \text{if } \tilde{\boldsymbol{x}}_t^\top \overline{\boldsymbol{\omega}}_t - \gamma > 0 \\ 0 & \text{if } \tilde{\boldsymbol{x}}_t^\top \overline{\boldsymbol{\omega}}_t - \gamma \leq 0. \end{cases}$
    $\boldsymbol{x}_t \leftarrow p_t \boldsymbol{x}^\diamond + (1 - p_t)\tilde{\boldsymbol{x}}_t$
    Play the SDMC by using strategy $\boldsymbol{x}_t$
    Observe terminal node $z_t$, utility $u_t$, and cost $c_t$
    Compute $\overline{\boldsymbol{\theta}}_{t+1}, \overline{\boldsymbol{\omega}}_{t+1}, \underline{\boldsymbol{\omega}}_{t+1}$ using $\delta' = \frac{\delta}{4T|\Sigma|}$
    $t \leftarrow t + 1$
  **end while**

The following theorem provides guarantees for Algorithm 2.

**Theorem 4.** *In repeated SDMCs satisfying Assumption 2, Algorithm 2 with $\delta \in (0,1)$ and $T \in \mathbb{N}_{>0}$ as inputs is $\delta$-safe and, with probability at least $1 - 2\delta$, the following holds:*

$$R_T \leq C + \frac{6}{\lambda}\Delta^2 |\Sigma| \sqrt{2T \log(2/\delta)},$$

*where $C$ is a term independent from $T$.*

Notice that requiring $\delta$-safeness introduces an extra $\Delta/\lambda$ factor multiplying the $\sqrt{T}$ term to the regret bound obtained in Theorem 2. Indeed, as we show in the following subsection, the dependence of the $\sqrt{T}$ term on $1/\lambda$ is necessary.

### 7.2. Lower Bound for the Hard–Threshold Problem

We conclude the section by showing that the regret bound attained by Algorithm 2 is asymptotically—in the parameter $T$—tight with respect to the parameters $\lambda$ and $T$.

**Theorem 5.** *In repeated SDMCs satisfying Assumption 2, if an algorithm is $\delta$-safe for any $\delta \in (0,1)$ given as input, then it incurs in a cumulative regret $R_T = \Omega(\frac{1}{\lambda}\sqrt{T})$ with probability at least $\frac{3}{4} - \delta$.*

## 8. Application to Sequential Games

In this section, we apply our algorithms in sequential game settings captured by the repeated SDMC model. In particular, in Section 8.1, we formulate the *utility-constrained opponent exploitation* problem introduced by Bernasconi-de-Luca et al. (2021) as an instance of our hard-threshold problem. Then, in Section 8.2, we experimentally evaluate our hard-threshold algorithm (Algorithm 2, abbreviated with HT) on a standard testbed of Poker-inspired sequential games, comparing them with the algorithm proposed by Bernasconi-de-Luca et al. (2021). Finally, we report additional experimental results evaluating our algorithm for the soft-threshold problem (Algorithm 1, abbreviated with ST) in Appendix E.

### 8.1. SDMCs for Constrained Opponent Exploitation

The tree structure underlying an SDMC can be easily shaped so as to represent the *two-player extensive-form games* studied by Bernasconi-de-Luca et al. (2021).[7] In particular, they study games in which one player, say Player 1, repeatedly faces an opponent, say Player 2, whose behavior follows a fixed, unknown stochastic strategy. The goal of Player 1 is to exploit the opponent (*i.e.*, maximizing their expected utility) while at the same time guaranteeing that the expected utility of the opponent remains above a given threshold.

Intuitively, the model by Bernasconi-de-Luca et al. (2021) naturally fits into our framework, as follows:

- Player 1's utilities are mapped to utilities of the SDMC, while Player 2's utilities corresponds to its costs.
- The information sets of Player 1 are mapped one-to-one to the decision nodes of the SDMC.
- The observation nodes of the SDMC are arranged so that their signals represent (stochastic) actions of Player 2 and chance that are observable by Player 1.
- The stochasticity of all the other unobservable actions (of both Player 2 and chance) is encoded into the random variables defining utilities and costs in the SDMC.

We refer the reader to Figure 1 for an example of SDMC resulting from a simple Poker game instance.

Notice that, in the work by Bernasconi-de-Luca et al. (2021), it is required that the constraint on the opponent's utility be satisfied at each round with high probability. This matches the requirement of our hard-threshold problem. Thus, in this case, we can directly compare the HT algorithm with the COX-UCB algorithm by Bernasconi-de-Luca et al. (2021).

**Differences between the HT algorithm and COX-UCB.** There is a crucial difference between our HT algorithm and COX-UCB. In particular, the latter requires perfect observability of the opponent's private information at the end of each episode. This significantly limits the applicability of the algorithm, as there are many games in which private information is *not* publicly revealed (*e.g.*, in Poker, when a player folds their private cards are not revealed to the other

[7] We refer the reader to the book by Shoham & Leyton-Brown (2008) for further details on extensive-form games.

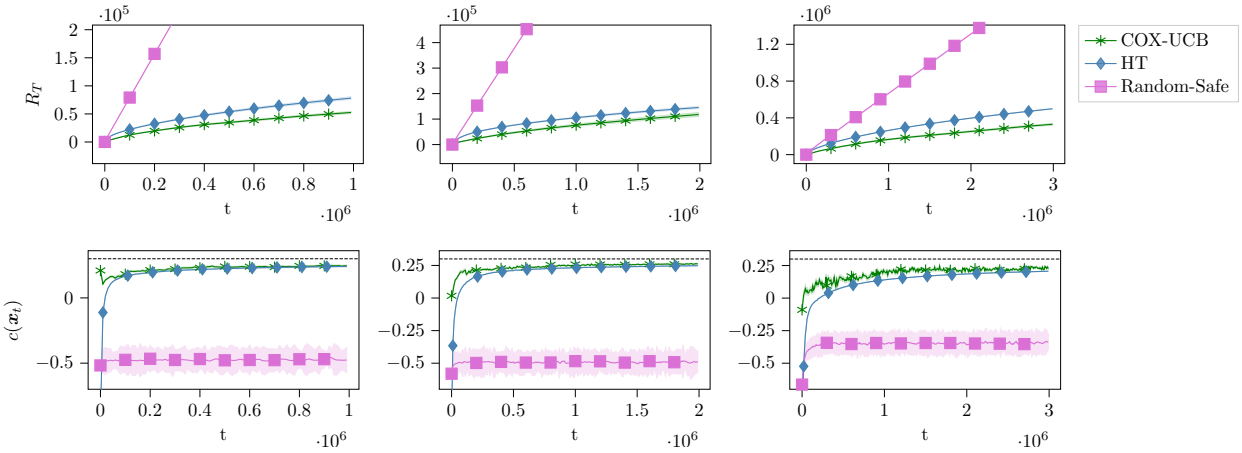


*Figure 3.* Experimental results for the hard-threshold problem. From left to right, the plots show the results for K5, K7, and L2 games.

players). On the other hand, our algorithm does *not* require knowledge of the opponent's private information, and, thus, it can be applied to a wider range of real-world sequential games than the COX-UCB algorithm.

### 8.2. Experiments for the Hard–Threshold Problem

We evaluate the algorithms on a testbed of Kuhn and Leduc Poker games, which is commonly used in the literature on sequential games.

**Setting.** We conduct experiments on Kuhn Poker with 5 and 7 ranks (called K5 and K7, respectively) and on Leduc Poker with 4 seeds and 2 ranks (called L2).[8] In order to adhere to the assumption on the existence of an always-strictly-safe strategy $\boldsymbol{x}^\diamond$ (Assumption 2), we enrich the games of Kuhn and Leduc Poker with a *forfeit* action causing the loss of the agent's bet without entering the game. Notice that a similar assumption is required by COX-UCB, as the optimization problem solved at each iteration by the algorithm needs to be always feasible (Bernasconi-de-Luca et al., 2021). In the experiments, we set $\gamma = 0.3$, while the values of $[\alpha, \beta]$ for COX-UCB utility constraints are set to $\alpha = -0.3$ and $\beta = +\infty$, respectively. All the optimization problems are solved by means of Gurobi (Gurobi Optimization, LLC, 2021). Moreover, we also compare the algorithms with an additional baseline (called Random-Safe, or RS for short), which selects a random strategy from a set of strategies that are safe with respect to the upper bounds on the costs, and, thus, safe with high probability. For further experimental results we refer the reader to Appendix E.

[8]Notice that these games are *zero-sum*, and, thus, requiring that the opponent's utility remains above a given threshold as in the framework by Bernasconi-de-Luca et al. (2021) is achieved in our model by letting the costs be equal to the opposite of the opponent's utility, which coincides with the agent's utility.

**Results.** We report in Figure 3 the results of running the HT, COX-UCB, and RS algorithms on K5, K7, and L2. As the plots show, our HT algorithm matches the performances (in terms of cumulative regret) of the COX-UCB algorithm. This is surprising since, as we discussed earlier, COX-UCB has much more information available than our algorithm. Indeed, this information advantage enables COX-UCB to incur in lower regret than HT in the first rounds, but then the cumulative regrets incurred by the two algorithms are comparable. Moreover, let us remark that, while the COX-UCB algorithm requires the solution of a bilinear optimization problem at each iteration, our HT algorithm only calls for the solution of an LP, thus requiring much less pre-round computational burden. These results elect our HT algorithm as the most appealing method to be applied in utility-constrained opponent exploitation problems. Finally, as shows in Figure 3, let us remark that our HT algorithm empirically satisfies the safety constraint at each round, thus validating our theoretical analysis.

## 9. Conclusions and Future Works

We studied, for the first time, safe learning in *tree-form sequential decision making* problems with *stochastic utilities* and *costs*. Our work paves the way to the application of tree-form sequential decision making to several real-world scenario, where being able to satisfy safety constraints is crucial, due to, *e.g.*, the presence of humans in the loop.

Future research could investigate other forms of safety, such as, *e.g.*, guaranteeing better performances with respect to a baseline (Garcelon et al., 2020; Yang et al., 2021; Bernasconi de Luca et al., 2021), ensuring monotonic improvements of the learned strategy (Garcıa & Fernández, 2015), and budget constraints (Badanidiyuru et al., 2013; Immorlica et al., 2019; Castiglioni et al., 2022).

## References


Amani, S., Alizadeh, M., and Thrampoulidis, C. Generalized linear bandits with safety constraints. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3562–3566. IEEE, 2020.

Badanidiyuru, A., Kleinberg, R., and Slivkins, A. Bandits with knapsacks. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science, FOCS 2013*, pp. 207–216. IEEE, 2013.

Bernasconi-de-Luca, M., Cacciamani, F., Fioravanti, S., Gatti, N., Marchesi, A., and Trovò, F. Exploiting opponents under utility constraints in sequential games. In *Advances in Neural Information Processing Systems*, 2021.

Bernasconi de Luca, M., Vittori, E., Trovò, F., and Restelli, M. Conservative online convex optimization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 19–34. Springer, 2021.

Brown, N. and Sandholm, T. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

Brown, N. and Sandholm, T. Superhuman ai for multiplayer poker. *Science*, 365(6456):885–890, 2019.

Castiglioni, M., Celli, A., and Kroer, C. Online learning with knapsacks: the best of both worlds. In *International Conference on Machine Learning*. PMLR, 2022.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge university press, 2006.

Chen, K., Cai, K., Huang, L., and Lui, J. C. Beyond the click-through rate: web link selection with multi-level feedback. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3308–3314, 2018.

Farina, G. and Sandholm, T. Model-free online learning in unknown sequential decision making problems and games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 5381–5390, 2021.

Farina, G., Schmucker, R., and Sandholm, T. Bandit linear optimization for sequential decision making and extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 5372–5380, 2021.

Garcelon, E., Ghavamzadeh, M., Lazaric, A., and Pirotta, M. Improved algorithms for conservative exploration in bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3962–3969, 2020.

Garcıa, J. and Fernández, F. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2021. URL https://www.gurobi.com.

Immorlica, N., Sankararaman, K. A., Schapire, R., and Slivkins, A. Adversarial bandits with knapsacks. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019*, pp. 202–219. IEEE Computer Society, 2019.

Koller, D., Megiddo, N., and von Stengel, B. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14:247–259, 1996.

Kuhn, H. W. 9. a simplified two-person poker. In *Contributions to the Theory of Games (AM-24), Volume I*, pp. 97–104. Princeton University Press, 2016.

Lattimore, T. and Szepesvári, C. *Bandit algorithms*. Cambridge University Press, 2020.

Liu, X., Li, B., Shi, P., and Ying, L. An efficient pessimistic-optimistic algorithm for stochastic linear bandits with general constraints. *Advances in Neural Information Processing Systems*, 34:24075–24086, 2021.

Pacchiano, A., Ghavamzadeh, M., Bartlett, P., and Jiang, H. Stochastic bandits with linear constraints. In *International Conference on Artificial Intelligence and Statistics*, pp. 2827–2835. PMLR, 2021.

Saxena, V., Jalden, J., and Gonzalez, J. Thompson sampling for linearly constrained bandits. In *International Conference on Artificial Intelligence and Statistics*, pp. 1999–2009. PMLR, 2020.

Shoham, Y. and Leyton-Brown, K. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Usmanova, I., Krause, A., and Kamgarpour, M. Safe convex learning under uncertain constraints. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2106–2114. PMLR, 2019.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

von Stengel, B. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, 1996.

Yang, Y., Wu, T., Zhong, H., Garcelon, E., Pirotta, M., Lazaric, A., Wang, L., and Du, S. S. A unified framework for conservative exploration. *arXiv preprint arXiv:2106.11692*, 2021.

# Appendix

The appendix includes all the proofs omitted from the paper and additional experimental results.

## A. Proofs Omitted from Section 4

**Theorem 1.** *In general repeated SDMCs, if an algorithm is $\delta$-safe for any $\delta \in (0,1)$ given as input, then it incurs in a regret $R_T = \Omega(T)$ with probability at least $1-\delta$.*

*Proof.* In order to prove the result, we consider two instances of SDMC, defined as follows. Both of them feature a single decision node of the agent, where there are 3 actions available, and no observation point. Thus, other than the empty sequence $\varnothing$, there are 3 different (one-action) sequences, namely $\Sigma = \{\varnothing, \sigma_1, \sigma_2, \sigma_\diamond\}$. The sequence $\sigma_\diamond$ is added only to guarantee that there always exists a strategy in $\mathcal{X}$ that satisfies the safety constraint. Instead, the other sequences characterize the two instances. In particular, instance $\mathrm{i}^{(i)}$ for $i \in \{1,2\}$ is specified as follows:

$$\mathrm{i}^{(i)} \coloneqq \begin{cases} \boldsymbol{\theta}^\star[\sigma_i] = \frac{1}{2},\ \boldsymbol{\omega}^\star[\sigma_i] = \frac{1}{2} \\ \boldsymbol{\theta}^\star[\sigma_j] = \frac{1}{2},\ \boldsymbol{\omega}^\star[\sigma_j] = \frac{1}{2} + \epsilon, \quad j \in \{1,2\} \setminus \{i\} \\ \boldsymbol{\theta}^\star[\sigma_\diamond] = 0,\ \boldsymbol{\omega}^\star[\sigma_\diamond] = \frac{1}{2}. \end{cases}$$

For all the instances we let $\gamma = \frac{1}{2}$. Notice that, in each instance, we can define $U_z$ and $C_z$ as Bernoulli random variables with suitable parameters, so that their expected values correctly define vectors $\boldsymbol{\theta}^\star$ and $\boldsymbol{\omega}^\star$. Moreover, in the rest of the proof, we denote by $\mathbb{P}^{(i)}$ the probability measure of instance $\mathrm{i}^{(i)}$, encompassing the randomization of both the SDMC and the algorithm (Lattimore & Szepesvári, 2020).

Any algorithm that is $\delta$-safe for any $\delta \in \left(0, \frac{1}{2}\right)$ given as input, when executed on instance $\mathrm{i}^{(i)}$, must output strategies $\boldsymbol{x}_t \in \mathcal{X}$ such that the following holds:

$$\mathbb{P}^{(i)}\Big\{ v_t \leq 0 \quad \forall t \in [T] \Big\} \geq 1 - \delta,$$

which implies that

$$\mathbb{P}^{(i)}\Big\{ \boldsymbol{x}_t[\sigma_i] + \boldsymbol{x}_t[\sigma_\diamond] = 1 \quad \forall t \in [T] \Big\} \geq 1 - \delta,$$

since the only strategies $\boldsymbol{x}_t$ such that $v_t = \boldsymbol{x}_t^\top \boldsymbol{\omega}^\star - \gamma \leq 0$ are those placing all the probability mass on sequences $\sigma_i$ and $\sigma_\diamond$, namely $\boldsymbol{x}_t[\sigma_i] + \boldsymbol{x}_t[\sigma_\diamond] = 1$.

Now, we relate the probability measure of instance $\mathrm{i}^{(1)}$ with that of instance $\mathrm{i}^{(2)}$ by means of the Pinsker's inequality (Cesa-Bianchi & Lugosi, 2006). Formally:

$$\begin{aligned} &\mathbb{P}^{(1)}\Big\{ \boldsymbol{x}_t[\sigma_2] + \boldsymbol{x}_t[\sigma_\diamond] = 1 \quad \forall t \in [T] \Big\} \\ &\quad \geq \mathbb{P}^{(2)}\Big\{ \boldsymbol{x}_t[\sigma_2] + \boldsymbol{x}_t[\sigma_\diamond] = 1 \quad \forall t \in [T] \Big\} - \sqrt{\frac{1}{2}\mathcal{K}(\mathbb{P}^{(2)}, \mathbb{P}^{(1)})}, \end{aligned}$$

where $\mathcal{K}(\mathbb{P}^{(2)}, \mathbb{P}^{(1)})$ is the Kullback-Leibler divergence between the probability measures of instance $\mathrm{i}^{(2)}$ and instance $\mathrm{i}^{(1)}$.

The Kullback-Leibler decomposition (see, *e.g.*, (Lattimore & Szepesvári, 2020) for more details) states that:

$$\mathcal{K}(\mathbb{P}^{(2)}, \mathbb{P}^{(1)}) \leq \epsilon^2 T.$$

Hence, we can conclude that:

$$\mathbb{P}^{(1)}\Big\{ \boldsymbol{x}_t[\sigma_2] + \boldsymbol{x}_t[\sigma_\diamond] = 1 \quad \forall t \in [T] \Big\} \geq 1 - \delta - \epsilon\sqrt{\frac{T}{2}}. \tag{2}$$

Let $R_T^{(i)}$ be the cumulative regret experienced by an algorithm in instance $\mathrm{i}^{(i)}$. It is easy to check that:

$$R_T^{(i)} = \frac{1}{2}\sum_{t=1}^{T} \boldsymbol{x}_t[\sigma_\diamond].$$

Then, under the probability measure of instance $\mathfrak{i}^{(1)}$, we have that:

$$\mathbb{P}^{(1)}\left\{R_T^{(1)} = \frac{T}{2} - \frac{1}{2}\sum_{t=1}^{T} \boldsymbol{x}_t[\sigma_1]\right\} \geq 1 - \delta.$$

Moreover, thanks to Equation (2), under the probability measure of the other instance $\mathfrak{i}^{(1)}$, the same holds for $R_T^{(2)}$ with probability at least $1 - \delta - \epsilon\sqrt{\frac{T}{2}}$, formally:

$$\mathbb{P}^{(1)}\left\{R_T^{(2)} = \frac{T}{2} - \frac{1}{2}\sum_{t=1}^{T} \boldsymbol{x}_t[\sigma_2]\right\} \geq 1 - \delta - \epsilon\sqrt{\frac{T}{2}}.$$

By a union bound, under the probability measure of instance $\mathfrak{i}^{(1)}$, we can conclude that the following holds with probability at least $1 - 2\delta - \epsilon\sqrt{\frac{T}{2}}$:

$$\begin{aligned}
2\max_{i\in\{1,2\}} R_T^{(i)} &\geq R_T^{(1)} + R_T^{(2)} \\
&= T - \frac{1}{2}\sum_{t=1}^{T}(\boldsymbol{x}_t[\sigma_1] + \boldsymbol{x}_t[\sigma_2]) \\
&= \frac{T}{2} + \frac{1}{2}\sum_{t=1}^{T}\boldsymbol{x}_t[\sigma_\diamond] \\
&\geq \frac{T}{2}.
\end{aligned}$$

The statement follows by setting $\epsilon = \delta\sqrt{\frac{2}{T}}$ and rescaling the parameter $\delta$ accordingly. □

## B. Proofs Omitted from Section 5

**Lemma 1.** *For any $t \in [T]$, $\mathbb{E}\left[\widehat{\boldsymbol{\theta}}_t\right] = \boldsymbol{\theta}^\star$ and $\mathbb{E}\left[\widehat{\boldsymbol{\omega}}_t\right] = \boldsymbol{\omega}^\star$.*

*Proof.* Fix $t \in [T]$. In order to prove the result, let us first notice that, for every $\sigma \in \Sigma$ and $\tau \in [t-1]$, it holds:

$$\begin{aligned}
\mathbb{E}\left[u_\tau \mathbb{I}\left\{\sigma(z_\tau) = \sigma\right\}\right] &= \boldsymbol{\theta}^\star[\sigma] \\
\mathbb{E}\left[c_\tau \mathbb{I}\left\{\sigma(z_\tau) = \sigma\right\}\right] &= \boldsymbol{\omega}^\star[\sigma].
\end{aligned}$$

Furthermore, for every $\sigma \in \Sigma$, we can write the following:

$$\sum_{\tau\in\tau_t^{ter}(\sigma)} u_\tau = \sum_{\tau\in\tau_t^{pure}(\sigma)} u_\tau \mathbb{I}\left\{\sigma(z_\tau) = \sigma\right\} \tag{3a}$$

$$\sum_{\tau\in\tau_t^{ter}(\sigma)} c_\tau = \sum_{\tau\in\tau_t^{pure}(\sigma)} c_\tau \mathbb{I}\left\{\sigma(z_\tau) = \sigma\right\}. \tag{3b}$$

Then, it follows that:

$$\begin{aligned}
\mathbb{E}\left[\widehat{\boldsymbol{\theta}}_t[\sigma]\right] &= \mathbb{E}\left[\frac{1}{n_t(\sigma)}\sum_{\tau\in\tau_t^{pure}(\sigma)} u_\tau \mathbb{I}\left\{\sigma(z_\tau) = \sigma\right\}\right] \\
&= \boldsymbol{\theta}^\star[\sigma],
\end{aligned}$$

and, similarly:

$$\mathbb{E}\left[\widehat{\boldsymbol{\omega}}_t[\sigma]\right] = \mathbb{E}\left[\frac{1}{n_t(\sigma)}\sum_{\tau\in\tau_t^{pure}(\sigma)} c_\tau \mathbb{I}\left\{\sigma(z_\tau) = \sigma\right\}\right]$$

$$= \boldsymbol{\omega}^\star[\sigma].$$

This concludes the proof. □

**Lemma 2.** *Given a confidence level $\delta' \in (0,1)$, for every $t \in [T]$ and $\sigma \in \Sigma$, the following bounds hold:*

$$\mathbb{P}\left\{\left|\widehat{\boldsymbol{\theta}}_t[\sigma] - \boldsymbol{\theta}^\star[\sigma]\right| \leq \boldsymbol{\xi}_t[\sigma]\right\} \geq 1-\delta',$$
$$\mathbb{P}\left\{\left|\widehat{\boldsymbol{\omega}}_t[\sigma] - \boldsymbol{\omega}^\star[\sigma]\right| \leq \boldsymbol{\xi}_t[\sigma]\right\} \geq 1-\delta',$$

*where $\boldsymbol{\xi}_t \in \mathbb{R}_{\geq 0}^{|\Sigma|}$ is a vector such that, for every $\sigma \in \Sigma$:*

$$\boldsymbol{\xi}_t[\sigma] \coloneqq \Delta\sqrt{\frac{\log(2/\delta')}{2n_t(\sigma)}}.$$

*Proof.* Fix any round $t \in [T]$ and sequence $\sigma \in \Sigma$. By the Hoeffding's inequality and Equation (3) in the proof of Lemma 1, we can write the following two equations:

$$\mathbb{P}\left\{\left|\widehat{\boldsymbol{\theta}}_t(\sigma) - \boldsymbol{\theta}^\star[\sigma]\right| \geq \frac{\ell}{n_t(\sigma)}\right\} \leq 2\exp\left(-\frac{2\ell^2}{n_t(\sigma)\Delta^2}\right)$$
$$\mathbb{P}\left\{\left|\widehat{\boldsymbol{\omega}}_t[\sigma] - \boldsymbol{\omega}^\star[\sigma]\right| \geq \frac{\ell}{n_t(\sigma)}\right\} \leq 2\exp\left(-\frac{2\ell^2}{n_t(\sigma)\Delta^2}\right).$$

By setting each right-hand-side of the equations above to be equal to $\delta'$ and solving each equation independently for $\ell$, we get the statement of the lemma. □

## C. Proofs Omitted from Section 6

**Lemma 3.** *The strategies $\boldsymbol{x}_t \in \mathcal{X}$ selected by Algorithm 1 with $\delta \in (0,1)$ and $T \in \mathbb{N}_{>0}$ as inputs are such that, with probability at least $1-\frac{\delta}{2}$, it holds:*

$$\sum_{t=1}^{T} \boldsymbol{x}_t^\top \boldsymbol{\xi}_t \leq 2\Delta|\Sigma|\sqrt{2T\log(2/\delta)}.$$

*Proof.* First, let us notice that, by considering the pure strategies $\boldsymbol{\pi}_t \in \Pi$ sampled according to the sequence-form strategies $\boldsymbol{x}_t$, we can write the following:

$$\sum_{t=1}^{T} \boldsymbol{\pi}_t^\top \boldsymbol{\xi}_t = \sum_{t=1}^{T}\sum_{\sigma\in\Sigma} \boldsymbol{\pi}_t[\sigma]\boldsymbol{\xi}_t[\sigma] \tag{4}$$
$$= \sum_{\sigma\in\Sigma}\sum_{t\in\tau_T^{pure}(\sigma)} \boldsymbol{\xi}_t[\sigma] \tag{5}$$
$$= \sum_{\sigma\in\Sigma}\sum_{t=1}^{n_T(\sigma)} \Delta\sqrt{\frac{\log(2/\delta)}{2t}} \tag{6}$$
$$\leq 2\Delta\sqrt{\frac{\log(2/\delta)}{2}}\sum_{\sigma\in\Sigma}\sqrt{n_T(\sigma)} \tag{7}$$
$$\leq \Delta|\Sigma|\sqrt{2\log(2/\delta)T}, \tag{8}$$

where Equation (5) holds by definition of $\tau_T^{pure}(\sigma)$, Equation (6) follows by using the definition of $\boldsymbol{\xi}_t[\sigma]$ and re-arranging the $\frac{1}{t}$ terms in the sum over $t \in \tau_T^{pure}(\sigma)$, Equation (7) holds since $\sum_{t=1}^{n_T(\sigma)} \frac{1}{\sqrt{t}} \leq 2\sqrt{n_T(\sigma)}$, while Equation (8) is obtained by Cauchy–Schwarz inequality.

Next, it remains to bound the difference between the sums $\sum_{t=1}^{T} \boldsymbol{x}_t^\top \boldsymbol{\xi}_t$ and $\sum_{t=1}^{T} \boldsymbol{\pi}_t^\top \boldsymbol{\xi}_t$. This can be done by applying the Azuma-Hoeffding inequality (Cesa-Bianchi & Lugosi, 2006). In particular, define the random variable $Z_t \coloneqq \sum_{\tau=1}^{t} \boldsymbol{\pi}_\tau^\top \boldsymbol{\xi}_\tau - \boldsymbol{x}_\tau^\top \boldsymbol{\xi}_\tau$ for every $t \in [T]$. It follows that $Z_t$ is a martingale, since the following holds:

$$\mathbb{E}\left[\boldsymbol{\pi}_t | \mathcal{F}_{t-1}\right] = \boldsymbol{x}_t,$$

where $\mathcal{F}_{t-1}$ is the filtration generated by the feedbacks observed by the algorithm up to round $t-1$. Hence, we can conclude that with probability at least $1-\delta'$ , it holds:

$$\sum_{t=1}^{T} \boldsymbol{x}_t^\top \boldsymbol{\xi}_t \leq \sum_{t=1}^{T} \boldsymbol{\pi}_t^\top \boldsymbol{\xi}_t + \Delta|\Sigma|\sqrt{2T \log\left(1/\delta'\right)}. \tag{9}$$

The lemma is proved by combining Equation (8) with Equation (9) obtained before, after setting $\delta' \coloneqq \frac{\delta}{2}$. □

**Theorem 2.** *In a general repeated SDMC, Algorithm 1 with $\delta \in (0,1)$ and $T \in \mathbb{N}_{>0}$ as inputs guarantees that, with probability at least $1-\delta$, the following bounds hold:*

$$R_T \leq 4\Delta|\Sigma|\sqrt{2T\log(2/\delta)}, \;\; V_T \leq 4\Delta|\Sigma|\sqrt{2T\log(2/\delta)}.$$

*Proof of Theorem 2.* As customary in the analysis of online learning algorithms, for every round $t \in [T]$, we define the following useful *clean events*:

$$\mathcal{E}_t^u \coloneqq \left\{\left|\widehat{\boldsymbol{\theta}}_t[\sigma] - \boldsymbol{\theta}^\star[\sigma]\right| \leq \boldsymbol{\xi}_t[\sigma] \quad \forall \sigma \in \Sigma\right\}$$
$$\mathcal{E}_t^c \coloneqq \left\{\left|\widehat{\boldsymbol{\omega}}_t[\sigma] - \boldsymbol{\omega}^\star[\sigma]\right| \leq \boldsymbol{\xi}_t[\sigma] \quad \forall \sigma \in \Sigma\right\},$$

Moreover, we let $\mathcal{E}_t \coloneqq \mathcal{E}_t^u \cap \mathcal{E}_t^c$. Notice that, by Lemma 2 and the fact that in Algorithm 1 the bounds $\overline{\boldsymbol{\theta}}_t = \widehat{\boldsymbol{\theta}}_t + \boldsymbol{\xi}_t$ and $\underline{\boldsymbol{\omega}}_t = \widehat{\boldsymbol{\omega}}_t - \boldsymbol{\xi}_t$ are computed by using confidence level $\delta' = \frac{\delta}{4T|\Sigma|}$, we can conclude that the event $\mathcal{E}_t^u$, respectively $\mathcal{E}_t^c$, holds jointly over $t \in [T]$, with probability at least $1 - \frac{\delta}{4}$. Thus, $\mathcal{E}_t$ is a high-probability event, since by a union bound we get that $\mathcal{E}_t$ holds with probability at least $1 - \frac{\delta}{2}$, jointly over $t \in [T]$.

**Feasibility.** As a first step, we prove that the linear program solved by Algorithm 1 at each round is feasible with high probability. At each round $t \in [T]$, we define $\mathcal{U}_t$ as the event in which LP$(\overline{\boldsymbol{\theta}}_t, \underline{\boldsymbol{\omega}}_t)$ solved by Algorithm 1 is feasible. We prove the result by showing that $\mathcal{E}_t \subset \mathcal{U}_t$ for every $t \in [T]$. Let $\boldsymbol{x}^\diamond$ be the known sequence-form strategy that always belongs to the set $\mathcal{X}^\star$ by assumption. Then, if the clean event $\mathcal{E}_t$ holds at round $t$, we can write the following:

$$\boldsymbol{x}^{\diamond,\top} \underline{\boldsymbol{\omega}}_t \leq \boldsymbol{x}^{\diamond,\top} \boldsymbol{\omega}^\star \leq \gamma,$$

where the first inequality holds since $\boldsymbol{x}^\diamond$ is a vector of non-negative entries and $\underline{\boldsymbol{\omega}}_t \preceq \boldsymbol{\omega}^\star$ under $\mathcal{E}_t$, while the second inequality holds since $\boldsymbol{x}^\diamond \in \mathcal{X}^\star$. This shows that $\boldsymbol{x}^\diamond$ is also feasible for LP$(\overline{\boldsymbol{\theta}}_t, \underline{\boldsymbol{\omega}}_t)$, proving that $\mathcal{E}_t \subset \mathcal{U}_t$.

**Regret Bound.** After having established that the event $\mathcal{U}_t$ holds with high probability at each round $t \in [T]$, we are now ready to prove that Algorithm 1 attains small cumulative regret. First, let us notice that, by letting $\boldsymbol{x}^\star \in \operatorname{argmax}_{\boldsymbol{x} \in \mathcal{X}^\star} \sum_{t=1}^{T} \boldsymbol{x}^\top \boldsymbol{\theta}^\star$ be an optimal agent's strategy among those satisfying the safety constraints, we have:

$$R_T = \sum_{t=1}^{T} (\boldsymbol{x}^\star - \boldsymbol{x}_t)^\top \boldsymbol{\theta}^\star.$$

At each round $t \in [T]$, under the clean event $\mathcal{E}_t$, we have $\boldsymbol{\theta}^\star + 2\boldsymbol{\xi}_t \succeq \overline{\boldsymbol{\theta}}_t$. Thus, since $\boldsymbol{x}_t$ is a vector of positive entries, we can write the following:

$$\boldsymbol{x}_t^\top (\boldsymbol{\theta}^\star + 2\boldsymbol{\xi}_t) \geq \boldsymbol{x}_t^\top \overline{\boldsymbol{\theta}}_t.$$

Moreover, since under $\mathcal{E}_t \subset \mathcal{U}_t$ the strategy $\boldsymbol{x}_t$ is indeed an optimal solution to LP$(\overline{\boldsymbol{\theta}}_t, \underline{\boldsymbol{\omega}}_t)$, we have that:

$$\boldsymbol{x}_t^\top \overline{\boldsymbol{\theta}}_t \geq \boldsymbol{x}^\top \overline{\boldsymbol{\theta}}_t,$$

for any strategy $\boldsymbol{x} \in \mathcal{X}$ such that $\boldsymbol{x}^\top \underline{\boldsymbol{\omega}}_t \leq \gamma$. These strategies include $\boldsymbol{x}^\star$, since, under the clean event $\mathcal{E}_t$, it is always the case that $\boldsymbol{x}^\top \underline{\boldsymbol{\omega}}_t \leq \gamma$ implies $\boldsymbol{x}^\top \boldsymbol{\omega}^\star \leq \gamma$. Finally, under $\mathcal{E}_t$, it also holds that $\boldsymbol{x}^{\star,\top} \overline{\boldsymbol{\theta}}_t \geq \boldsymbol{x}^{\star,\top} \boldsymbol{\theta}^\star$. Thus, by combining all these observations we obtain that:

$$\boldsymbol{x}_t^\top (\boldsymbol{\theta}^\star + 2\boldsymbol{\xi}_t) \geq \boldsymbol{x}^{\star,\top} \boldsymbol{\theta}^\star,$$

which, after re-arranging, gives $2\boldsymbol{x}_t^\top \boldsymbol{\xi}_t \geq (\boldsymbol{x}^\star - \boldsymbol{x}_t)^\top \boldsymbol{\theta}^\star$. By Lemma 3 and the fact that $\mathcal{E}_t$ holds with probability at least $1 - \frac{\delta}{2}$ for every round $t \in [T]$, a union bound allows us to conclude that with probability at least $1 - \delta$:

$$\sum_{t=1}^{T} (\boldsymbol{x}^\star - \boldsymbol{x}_t)^\top \boldsymbol{\theta}^\star \leq 4\Delta|\Sigma|\sqrt{2T \log(2/\delta)},$$

which gives the desired regret bound.

**Violation Bound.** The bound is proved in a way similar to that followed for the regret. First, let us recall that, by definition of cumulative violation:

$$V_T = \sum_{t=1}^{T} (\boldsymbol{x}_t^\top \boldsymbol{\omega}^\star - \gamma)^+.$$

At each round $t \in [T]$, under the clean event $\mathcal{E}_t$ we have that $\boldsymbol{\omega}^\star - 2\boldsymbol{\xi}_t \preceq \underline{\boldsymbol{\omega}}_t$ and that $\mathrm{LP}(\overline{\boldsymbol{\theta}}_t, \underline{\boldsymbol{\omega}}_t)$ is feasible. Hence, since $\boldsymbol{x}_t$ has positive entries, we can state that:

$$\boldsymbol{x}_t^\top (\boldsymbol{\omega}^\star - 2\boldsymbol{\xi}_t) \leq \boldsymbol{x}_t^\top \underline{\boldsymbol{\omega}}_t \leq \gamma.$$

By re-arranging the terms, we have that $\boldsymbol{x}_t^\top \boldsymbol{\omega}^\star \leq \gamma + 2\boldsymbol{x}_t^\top \boldsymbol{\xi}_t$. This implies that $(\boldsymbol{x}_t^\top \boldsymbol{\omega}^\star - \gamma)^+ \leq 2\boldsymbol{x}_t^\top \boldsymbol{\xi}_t$. Finally, by Lemma 3 and the fact that $\mathcal{E}_t$ holds with probability at least $1 - \frac{\delta}{2}$ for every round $t \in [T]$, we can conclude that with probability at least $1 - \delta$:

$$V_T \leq 2 \sum_{t=1}^{T} \boldsymbol{x}_t^\top \boldsymbol{\xi}_t \leq 4\Delta|\Sigma|\sqrt{2T \log(2/\delta)},$$

which is obtained by a union bound. □

**Theorem 3.** *In general repeated SDMCs, if an algorithm guarantees $V_T = o(\sqrt{T})$ with probability at least $1 - \delta$ for any $\delta \in (0, \frac{1}{3})$ given as input, then it incurs in a regret $R_T = \Omega(T)$ with probability at least $1 - 3\delta$.*

*Proof.* We employ the same instances used to prove Theorem 1. In each instance $\mathfrak{i}^{(i)}$, the cumulative deviation is:

$$V_T^{(i)} = \epsilon \sum_{t=1}^{T} \boldsymbol{x}_t[\sigma_j],$$

where we let $j = 2$ when $i = 1$ and *vice versa*. Moreover, given an algorithm that guarantees $V_T = o(\sqrt{T})$ with probability at least $1 - \delta$ for any $\delta \in (0, 1)$ given as input, there must exist a constant $c > 0$ such that, for each instance $\mathfrak{i}^{(i)}$, under the probability measure of that instance the following hods with probability al least $1 - \delta$:

$$V_T^{(i)} \leq c\sqrt{T}. \tag{10}$$

Next, we express the event $V_T^{(1)} \leq c\sqrt{T}$ under the probability measure of instance $\mathfrak{i}^{(2)}$. This is done by means of the Pinsker's inequality. As in the proof of Theorem 1, we can conclude that, under the probability measure of instance $\mathfrak{i}^{(2)}$, the following holds with probability at least $1 - \delta - \epsilon\sqrt{\frac{T}{2}}$:

$$\sum_{t=1}^{T} \boldsymbol{x}_t[\sigma_2] \leq \frac{c}{\epsilon}\sqrt{T}. \tag{11}$$

Moreover, by Equation (10) and definition of $V_T^{(2)}$, we have:

$$\sum_{t=1}^{T} \boldsymbol{x}_t[\sigma_1] \leq \frac{c}{\epsilon}\sqrt{T}, \tag{12}$$

which holds with probability at least $1-\delta$ under the probability measure of instance $\mathfrak{i}^{(2)}$.

Now, let us consider the regret experienced by any algorithm in instance $\mathfrak{i}^{(2)}$, which is equal to $R_T^{(2)} = \frac{1}{2}\sum_{t=1}^{T} \boldsymbol{x}_t[\sigma_\diamond]$. Then, by a union bound and Equations (11) and (12), under the probability measure of instance $\mathfrak{i}^{(2)}$ we have that the following holds with probability at least $1-2\delta-\epsilon\sqrt{\frac{T}{2}}$:

$$\begin{aligned} R_T^{(2)} &= \frac{1}{2}\sum_{t=1}^{T} \boldsymbol{x}_t[\sigma_\diamond] \\ &= \frac{T}{2} - \sum_{t=1}^{T} \left(\boldsymbol{x}_t[\sigma_1] + \boldsymbol{x}_t[\sigma_2]\right) \\ &\geq \frac{T}{2} - \frac{2c}{\epsilon}\sqrt{\frac{T}{2}}. \end{aligned}$$

Then, setting $\epsilon = \frac{d}{\sqrt{2T}}$ gives:

$$R_T^{(2)} \geq T\left(\frac{1}{2} - \frac{2c}{d}\right)$$

with probability at least $1-2\delta-d$. Thus, by setting $d=\delta$ and $c=\frac{\delta}{8}$ we can conclude that $R_T^{(2)} \geq \frac{T}{4}$ with probability at least $1-3\delta$. This concludes the proof. □

## D. Proof Omitted from Section 7

**Theorem 4.** *In repeated SDMCs satisfying Assumption 2, Algorithm 2 with $\delta \in (0,1)$ and $T \in \mathbb{N}_{>0}$ as inputs is $\delta$-safe and, with probability at least $1-2\delta$, the following holds:*

$$R_T \leq C + \frac{6}{\lambda}\Delta^2|\Sigma|\sqrt{2T\log(2/\delta)},$$

*where $C$ is a term independent from $T$.*

*Proof.* Let us define the clean event $\mathcal{E}_t$ as in the proof of Theorem 2.

**Violations.** As a first step, we show that Algorithm 2 is $\delta$-safe for any $\delta \in (0,1)$ given as input. In order to prove that, it is enough to show that $v_t \leq 0$ for every $t \in [T]$, with probability at least $1-\delta$. In particular, under the event $\mathcal{E}_t$, we have that:

$$\begin{aligned} \boldsymbol{x}_t^\top \boldsymbol{\omega}^\star &= p_t \boldsymbol{x}^{\diamond,\top}\boldsymbol{\omega}^\star + (1-p_t)\tilde{\boldsymbol{x}}_t^\top \boldsymbol{\omega}^\star && (13)\\ &= p_t(\gamma-\lambda) + (1-p_t)\tilde{\boldsymbol{x}}_t^\top \boldsymbol{\omega}^\star && (14)\\ &\leq p_t(\gamma-\lambda) + (1-p_t)\tilde{\boldsymbol{x}}_t^\top \overline{\boldsymbol{\omega}}_t && (15)\\ &= p_t(\gamma-\lambda-\tilde{\boldsymbol{x}}_t^\top \overline{\boldsymbol{\omega}}_t) + \tilde{\boldsymbol{x}}_t^\top \overline{\boldsymbol{\omega}}_t && (16)\\ &\leq \gamma, && (17) \end{aligned}$$

where Equation (14) follows from the definition of $\boldsymbol{x}^\diamond$, Equation (15) comes from the fact that $\tilde{\boldsymbol{x}}_t$ has non-negative entries and, under the event $\mathcal{E}_t$, it holds $\boldsymbol{\omega}^\star \preceq \overline{\boldsymbol{\omega}}_t$, while Equation (17) follows from the definition of $p_t$. Hence, we can conclude that, under the clean event $\mathcal{E}_t$, it holds $v_t \leq 0$. Since the clean event $\mathcal{E}_t$ holds at every round $t \in [T]$ with probability at least $1-\frac{\delta}{2}$ (Lemma 2) and the fact that $\delta' = \frac{\delta}{4T|\Sigma|}$, we get that Algorithm 2 is $\delta$-safe for any $\delta \in (0,1)$ given as input.

**Regret Bound.** In order to prove that Algorithm 2 attains sublinear regret with high probability, we decompose its cumulative regret in the following way:

$$R_T = \sum_{t=1}^{T}(\boldsymbol{x}^\star - \boldsymbol{x}_t)^\top \boldsymbol{\theta}^\star = \sum_{t=1}^{T}(1-p_t)(\boldsymbol{x}^\star - \tilde{\boldsymbol{x}}_t)^\top \boldsymbol{\theta}^\star + p_t(\boldsymbol{x}^\star - \boldsymbol{x}^\diamond)^\top \boldsymbol{\theta}^\star$$

$$\leq \sum_{t=1}^{T}(1-p_t)(\boldsymbol{x}^\star - \tilde{\boldsymbol{x}}_t)^\top \boldsymbol{\theta}^\star + 2\Delta \sum_{t=1}^{T} p_t, \tag{18}$$

where the last step follows from the triangular inequality.

**Bounding $\sum_{t=1}^{T}(1-p_t)(\boldsymbol{x}^\star - \tilde{\boldsymbol{x}}_t)^\top \boldsymbol{\theta}^\star$.** Next, we bound the first term of Equation (18). Given that the strategy $\tilde{\boldsymbol{x}}_t$ is chosen analogously to the strategy $\boldsymbol{x}_t$ in Algorithm 1, we can follow steps analogous to those used in the proof of Theorem 2 in order to derive the following bound under the clean event $\mathcal{E}_t$:

$$(\boldsymbol{x}^\star - \tilde{\boldsymbol{x}}_t)^\top \boldsymbol{\theta}^\star \leq 2\tilde{\boldsymbol{x}}_t^\top \boldsymbol{\xi}_t.$$

Notice that, in this case, we cannot directly apply Lemma 3 as we no longer have that $\mathbb{E}[\boldsymbol{\pi}_t|\mathcal{F}_{t-1}] = \tilde{\boldsymbol{x}}_t$ (since the strategy actually played by the algorithm is $\boldsymbol{x}_t$ instead of $\tilde{\boldsymbol{x}}_t$). However, we obtain a similar result by decomposing the strategy $\boldsymbol{x}_t$ chosen by Algorithm 2. Indeed, from Lemma 3, we have that, with probability at least $1 - \frac{\delta}{2}$:

$$\sum_{t=1}^{T} \boldsymbol{x}_t^\top \boldsymbol{\xi}_t \leq 2\Delta|\Sigma|\sqrt{2T\log(2/\delta)},$$

since Lemma 3 does not rely on any assumption on how the strategies $\boldsymbol{x}_t$ are selected. Thus, we can decompose $\boldsymbol{x}_t$ in $p_t\boldsymbol{x}^\diamond + (1-p_t)\tilde{\boldsymbol{x}}_t$ and obtain that, with probability at least $1 - \frac{\delta}{2}$:

$$\sum_{t=1}^{T}(1-p_t)\tilde{\boldsymbol{x}}_t^\top \boldsymbol{\xi}_t \leq 2\Delta|\Sigma|\sqrt{2T\log(2/\delta)}, \tag{19}$$

where we discarded the sum of the positive terms $p_t\boldsymbol{x}^{\diamond,\top}\boldsymbol{\xi}_t$.

**Bounding $\sum_{t=1}^{T} p_t$.** Now, we bound the second term of Equation (18). In order to do that, we split the rounds into two sets. The first set $T_1$ is made by all the rounds $t \in [T]$ such that $p_t \leq \frac{1}{2}$, while the second set $T_2$ by all the rounds $t \in [T]$ such that $p_t > \frac{1}{2}$. In the following, we analyze the two cases separately.

(i) *Upper bound on $\sum_{t\in T_1} p_t$.* Notice that, under the clean event $\mathcal{E}_t$:

$$\begin{aligned} p_t &= \frac{\min\{\tilde{\boldsymbol{x}}_t^\top \overline{\boldsymbol{\omega}}_t, \Delta\} - \gamma}{\min\{\tilde{\boldsymbol{x}}_t^\top \overline{\boldsymbol{\omega}}_t, \Delta\} - \gamma + \lambda} \\ &\leq \frac{\tilde{\boldsymbol{x}}_t^\top \overline{\boldsymbol{\omega}}_t - \gamma}{\tilde{\boldsymbol{x}}_t^\top \overline{\boldsymbol{\omega}}_t - \gamma + \lambda} \\ &\leq \frac{\tilde{\boldsymbol{x}}_t^\top \overline{\boldsymbol{\omega}}_t - \gamma}{\lambda} \\ &\leq \frac{2}{\lambda}\tilde{\boldsymbol{x}}_t^\top \boldsymbol{\xi}_t, \end{aligned}$$

where the last inequality follows by $\tilde{\boldsymbol{x}}_t^\top \underline{\boldsymbol{\omega}}_t \leq \gamma$ and $\tilde{\boldsymbol{x}}_t^\top(\overline{\boldsymbol{\omega}}_t - \underline{\boldsymbol{\omega}}_t) = 2\tilde{\boldsymbol{x}}_t^\top \boldsymbol{\xi}_t$. By a union bound and recalling that the clean event $\mathcal{E}_t$ holds at every round $t \in [T]$ with probability at least $1 - \frac{\delta}{2}$, we can conclude that the following holds with probability at least $1 - \delta$:

$$\sum_{t\in T_1} p_t \leq \sum_{t\in T_1} \frac{2}{\lambda}\tilde{\boldsymbol{x}}_t^\top \boldsymbol{\xi}_t \leq \frac{4}{\lambda}\sum_{t\in T_1}(1-p_t)\tilde{\boldsymbol{x}}_t^\top \boldsymbol{\xi}_t \leq \frac{4}{\lambda}\sum_{t\in[T]}(1-p_t)\tilde{\boldsymbol{x}}_t^\top \boldsymbol{\xi}_t \leq \frac{4}{\lambda}\Delta|\Sigma|\sqrt{2T\log(2/\delta)}, \tag{20}$$

where the second inequality comes from $1 - p_t > \frac{1}{2}$ for $p_t < \frac{1}{2}$ and the last one by Equation (19).

(ii) *Upper bound on $\sum_{t\in T_2} p_t$.* We proceed by upper bounding the cardinality $|T_2|$ of the set $T_2$. In order to do that, we need to introduce the following set $T_3 \subset [T]$, which is defined as:

$$T_3 \coloneqq \left\{ t \in [T] \mid \boldsymbol{\pi}_t^\top \boldsymbol{\xi}_t \geq \frac{\lambda^2}{8\delta} \right\}.$$

By resorting to arguments similar to the ones used to prove Lemma 3, we can bound the range of the random variable $T_3$. In particular, let us consider the following steps, similar to the ones used in the proof of Lemma 3:

$$\sum_{t\in T_3} \boldsymbol{\pi}_t^\top \boldsymbol{\xi}_t = \sum_{t\in T_3}\sum_{\sigma\in\Sigma} \boldsymbol{\pi}_t[\sigma]\boldsymbol{\xi}_t[\sigma] \tag{21}$$

$$= \sum_{\sigma\in\Sigma} \sum_{t\in\tau_T^{pure}(\sigma)\cap T_3} \boldsymbol{\xi}_t[\sigma] \tag{22}$$

$$\leq \sum_{\sigma\in\Sigma} \sum_{t=1}^{|\tau_T^{pure}(\sigma)\cap T_3|} \Delta\sqrt{\frac{\log(2/\delta)}{2t}} \tag{23}$$

$$\leq 2\Delta\sqrt{\frac{\log(2/\delta)}{2}} \sum_{\sigma\in\Sigma} \sqrt{|\tau_T^{pure}(\sigma)\cap T_3|} \tag{24}$$

$$\leq \Delta|\Sigma|\sqrt{2|T_3|\log(2/\delta)}, \tag{25}$$

where the only difference with respect to the proof of Lemma 3 is in Equation (23), which uses the fact that $T_3$ is a subset of the set $[T]$ and, thus, for each sequence $\sigma\in\Sigma$ the $t$-th term in the sum in Equation (23) is an upper bound for the $t$-th term $\boldsymbol{\xi}_t[\sigma]$ in the sum in Equation (22). Then, since for all $t\in T_3$ we have that $\boldsymbol{\pi}_t^\top\boldsymbol{\xi}_t \geq \frac{\lambda^2}{8\Delta}$, it holds that $\sum_{t\in T_3}\boldsymbol{\pi}_t^\top\boldsymbol{\xi}_t \geq |T_3|\frac{\lambda^2}{8\Delta}$. This, together with Equation (25), implies that the following upper bound on $|T_3|$ holds:

$$|T_3| \leq 128\frac{\Delta^4|\Sigma|^2}{\lambda^2}\log(2/\delta).$$

By using the bound on $|T_3|$ given above, we can obtain an upper bound on $|T_2|$, as follows. First, let us observe that, if $t\in T_2$, then $2(\tilde{\boldsymbol{x}}_t^\top\overline{\boldsymbol{\omega}}_t - \gamma) \geq \tilde{\boldsymbol{x}}_t^\top\overline{\boldsymbol{\omega}}_t - \gamma + \lambda$. By rearranging the terms, we have that $\tilde{\boldsymbol{x}}_t^\top\overline{\boldsymbol{\omega}}_t \geq \gamma + \lambda$. Moreover, by rewriting the upper confidence bound $\overline{\boldsymbol{\omega}}_t$ as $\underline{\boldsymbol{\omega}}_t + 2\boldsymbol{\xi}_t$, we obtain that, if $t\in T_2$, then $\tilde{\boldsymbol{x}}_t^\top\boldsymbol{\xi}_t \geq \frac{\lambda}{2}$. By using the reverse Markov inequality (Lattimore & Szepesvári, 2020), we can lower bound the probability that $t\in T_2\cap T_3$. Indeed, if $t\in T_2$, then we can lower bound the probability that $\boldsymbol{\pi}_t^\top\boldsymbol{\xi}_t \geq \frac{\lambda^2}{8\Delta}$ holds. First, we lower bound the expected value of $\boldsymbol{\pi}_t^\top\boldsymbol{\xi}_t$ as follows (recall that $\mathcal{F}_{t-1}$ is the filtration generated by the information up to round $t-1$):

$$\mathbb{E}[\boldsymbol{\pi}_t^\top\boldsymbol{\xi}_t|\mathcal{F}_{t-1}] = \boldsymbol{x}_t^\top\boldsymbol{\xi}_t$$

$$= p_t\boldsymbol{x}^{\diamond,\top}\boldsymbol{\xi}_t + (1-p_t)\tilde{\boldsymbol{x}}_t^\top\boldsymbol{\xi}_t$$

$$\geq (1-p_t)\tilde{\boldsymbol{x}}_t^\top\boldsymbol{\xi}_t \tag{26}$$

$$\geq \frac{\lambda}{\Delta+\lambda}\tilde{\boldsymbol{x}}_t^\top\boldsymbol{\xi}_t \tag{27}$$

$$\geq \frac{\lambda^2}{4\Delta}, \tag{28}$$

where Equation (26) follows from the fact that $p_t\boldsymbol{x}^{\diamond,\top}\boldsymbol{\xi}_t \geq 0$, Equation (27) follows from the fact that $p_t \leq \frac{\Delta}{\Delta+\lambda}$ by definition of $p_t$ (since $x\mapsto\frac{x}{x+a}$ is monotonically increasing for $a>0$ and $\min\{\tilde{\boldsymbol{x}}_t^\top\overline{\boldsymbol{\omega}}_t,\Delta\}-\gamma\leq\Delta$), while Equation (28) follows from the fact that $\lambda<\Delta$ and that, if $t\in T_2$, then $\tilde{\boldsymbol{x}}_t^\top\boldsymbol{\xi}_t \geq \frac{\lambda}{2}$. Then, by the reverse Markov inequality we have that:

$$\mathbb{P}\left\{\boldsymbol{\pi}_t^\top\boldsymbol{\xi}_t \geq \frac{\lambda^2}{8\Delta} \mid \mathcal{F}_{t-1}\right\} \geq \frac{\mathbb{E}[\boldsymbol{\pi}_t^\top\boldsymbol{\xi}_t|\mathcal{F}_{t-1}] - \frac{\lambda^2}{8\Delta}}{\Delta|\Sigma| - \frac{\lambda^2}{8\Delta}} \geq \frac{\lambda^2}{4\Delta^2|\Sigma|}.$$

Let us define $\rho \coloneqq \frac{1}{4\Delta^2|\Sigma|}$, so that $\mathbb{P}\left\{\boldsymbol{\pi}_t^\top\boldsymbol{\xi}_t \geq \frac{\lambda^2}{8\Delta}\middle|\mathcal{F}_{t-1}\right\} \geq \rho\lambda^2$. Starting from this inequality we can now derive an upper bound on $|T_2|$. Suppose by contradiction that $|T_2| \geq (1+\alpha)128\frac{\Delta^4|\Sigma|^2}{\rho\lambda^4}\log(2/\delta)$, for a small $\alpha>0$. This would imply that $\mathbb{E}\Big[|T_3|\Big] \geq \rho\lambda^2|T_2| \geq (1+\alpha)128\frac{\Delta^4|\Sigma|^2}{\lambda^2}\log(2/\delta)$, contradicting the fact that $|T_3| \leq 128\frac{\Delta^4|\Sigma|^2}{\lambda^2}\log(2/\delta)$. Thus, by setting $\alpha=\frac{1}{128}$ for convenience, we obtain:

$$|T_2| \leq 129\frac{\Delta^4|\Sigma|^2}{\rho\lambda^4}\log(2/\delta). \tag{29}$$

Finally, by combining Equation (20) and Equation (29), the following holds with probability at least $1-\delta$:

$$\sum_{t=1}^{T} p_t \leq 516\frac{\Delta^6|\Sigma|^3}{\lambda^4}\log(2/\delta) + \frac{4}{\lambda}\Delta|\Sigma|\sqrt{2T\log(2/\delta)}.$$

**Putting all Together.** By combining the upper bound on $\sum_{t=1}^{T} p_t$ with Equation (18) and Equation (19), we can conclude that the following bound on the regret holds with probability at least $1-2\delta$:

$$R_T \leq 516\frac{\Delta^7|\Sigma|^3}{\lambda^4}\log(2/\delta) + \frac{6}{\lambda}\Delta^2|\Sigma|\sqrt{2T\log(2/\delta)}.$$

By letting $C \coloneqq 516\frac{\Delta^7|\Sigma|^3}{\lambda^4}\log(2/\delta)$, we conclude the proof. □

**Theorem 5.** *In repeated SDMCs satisfying Assumption 2, if an algorithm is $\delta$-safe for any $\delta \in (0,1)$ given as input, then it incurs in a cumulative regret $R_T = \Omega(\frac{1}{\lambda}\sqrt{T})$ with probability at least $\frac{3}{4}-\delta$.*

*Proof.* In order to prove the statement, we provide two instances of SDMC such that, if an algorithm is $\delta$-safe for any $\delta \in (0,1)$ given as input in the first instance, then the regret attained by the algorithm is at least $\frac{1}{8\lambda}\sqrt{T}$ in the second one. Let $\epsilon \in \mathbb{R}_+$ be a parameter to be defined later. We consider two instances $\mathfrak{i}^{(1)}$ and $\mathfrak{i}^{(2)}$, where there is only one decision node of the agent and no observation nodes, so that both instances have 3 sequences: the empty sequence $\varnothing$, the alway-strictly-safe sequence $\sigma_\diamond$, and an additional sequence $\sigma_1$. Hence, $\Sigma = \{\varnothing, \sigma_1, \sigma_\diamond\}$. The two instances deffer in the utilities and costs that are associated to sequence $\sigma_1$. More specifically, each instance $\mathfrak{i}^{(i)}$ for $i \in \{1,2\}$ is specified as follows:

$$\mathfrak{i}^{(i)} \coloneqq \begin{cases} \boldsymbol{\theta}^\star[\sigma_1] = \boldsymbol{\omega}^\star[\sigma_1] = \frac{1}{2}+\epsilon & \text{if } i=1 \\ \boldsymbol{\theta}^\star[\sigma_1] = \boldsymbol{\omega}^\star[\sigma_1] = \frac{1}{2} & \text{if } i=2 \\ \boldsymbol{\theta}^\star[\sigma_\diamond] = 0,\ \boldsymbol{\omega}^\star[\sigma_\diamond] = \frac{1}{2}-\lambda. \end{cases}$$

Finally, in both instances we set $\gamma = \frac{1}{2}$. Notice that it is always possible to define $U_z$ and $C_z$ as Bernoulli random variables whose expected values result in vectors $\boldsymbol{\theta}^\star$ and $\boldsymbol{\omega}^\star$ as above.

If an algorithm is $\delta$-safe on instance $\mathfrak{i}^{(1)}$, then:

$$\mathbb{P}^{(1)}\left\{\boldsymbol{x}_t[\sigma_1] \geq \frac{\epsilon}{\epsilon+\lambda} \quad \forall t \in [T]\right\} \geq 1-\delta,$$

where $\mathbb{P}^{(i)}$ is the probability measure of instance $\mathfrak{i}^{(i)}$. This implies the following:

$$\mathbb{P}^{(1)}\left\{\sum_{t=1}^{T}\boldsymbol{x}_t[\sigma_1] \geq T\frac{\epsilon}{\epsilon+\lambda} \quad \forall t \in [T]\right\} \geq 1-\delta.$$

Now, let us change the probability measure from $\mathbb{P}^{(1)}$ to $\mathbb{P}^{(2)}$, by means of the Pinsker's inequality:

$$\begin{aligned}&\mathbb{P}^{(2)}\left\{\sum_{t=1}^{T}\boldsymbol{x}_t[\sigma_1] \geq T\frac{\epsilon}{\epsilon+\lambda} \quad \forall t \in [T]\right\} \\ \geq&\mathbb{P}^{(1)}\left\{\sum_{t=1}^{T}\boldsymbol{x}_t[\sigma_1] \geq T\frac{\epsilon}{\epsilon+\lambda} \quad \forall t \in [T]\right\} - \sqrt{\frac{1}{2}\mathcal{K}(2,1)},\end{aligned}$$

where $\mathcal{K}(2,1)$ is the Kullback-Leibler divergence between the probabilities measures $\mathbb{P}^{(2)}$ and $\mathbb{P}^{(1)}$. Since standard computations show that $\mathcal{K}(2,1) \leq \epsilon\sqrt{\frac{T}{2}}$, we conclude that:

$$\mathbb{P}^{(2)}\left\{\sum_{t=1}^{T}\boldsymbol{x}_t[\sigma_1] \geq T\frac{\epsilon}{\epsilon+\lambda} \quad \forall t \in [T]\right\} \geq 1-\delta-\epsilon\sqrt{\frac{T}{2}}. \tag{30}$$

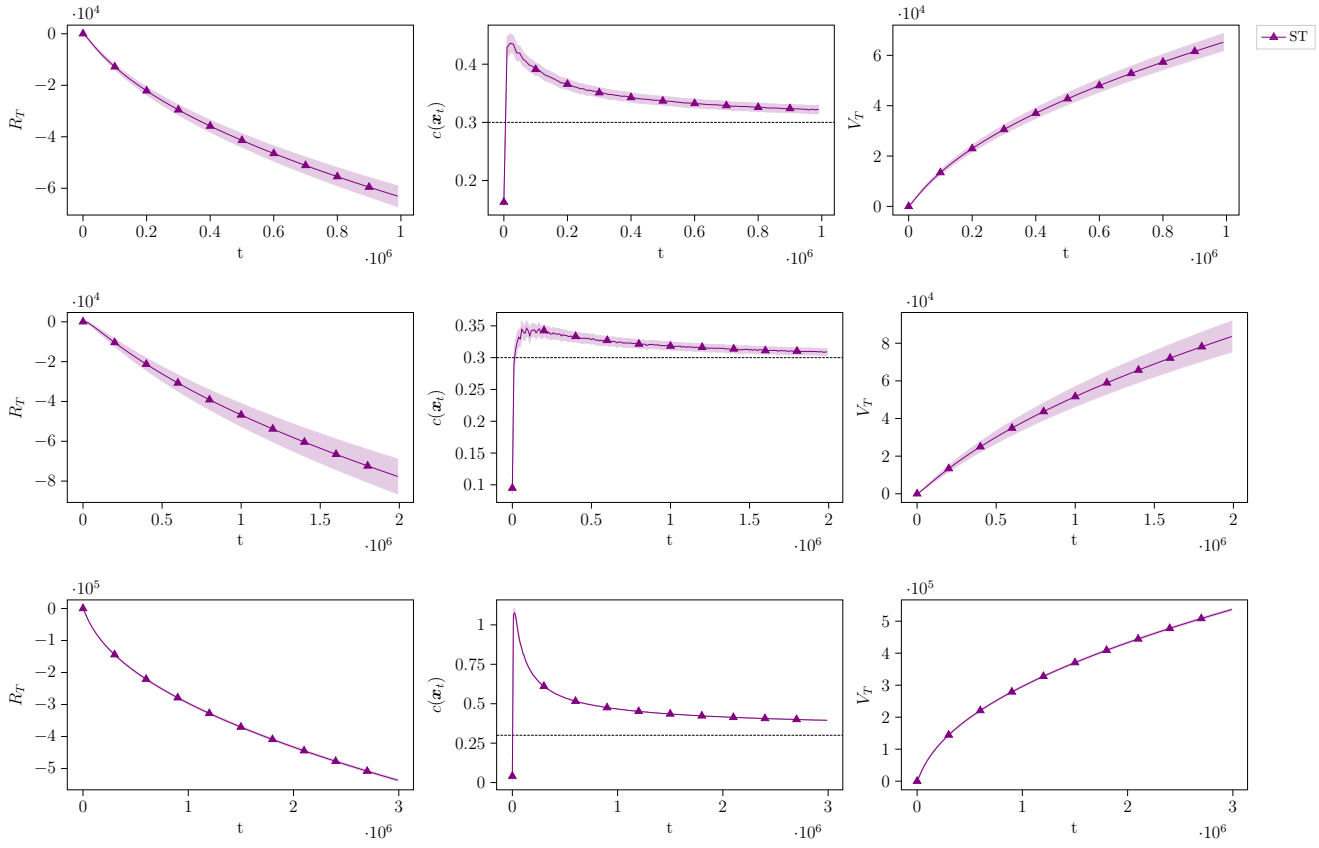


*Figure 4.* Experimental results for the soft-threshold problem on K5 (top row), K7 (center row) and L2 (bottom row). The plots show cumulative regret (left column), expected cost (center column) and cumulative violation (right column).

Next, let us consider the cumulative regret $R_T^{(2)}$ of the algorithm on instance $\mathfrak{i}^{(2)}$. It is easy to check that:

$$R_T^{(2)} = \sum_{t=1}^{T} \boldsymbol{x}_t[\sigma_1].$$

By setting $\epsilon = \frac{1}{4}\sqrt{\frac{2}{T}}$ and using Equation (30), we obtain that, under the probability measure of instance $\mathfrak{i}^{(2)}$, it holds $R_T \geq \frac{1}{8\lambda}\sqrt{2T}$ with probability at least $\frac{3}{4} - \delta$, concluding the proof. □

## E. Additional Experimental Results

In this section, we present additional experimental results on the soft-threshold problem.

Figure 4 shows the performances of the soft-threshold algorithm (Algorithm 1) in three different instances of Kuhn and Leduc Poker: Kuhn Poker with 5 ranks (K5), Kuhn Poker with 7 ranks (K7) and Leduc Poker with 4 seeds and 2 ranks (L2). The plots show the cumulative violation $V_T$, the cumulative regret $R_T$, and the expected cost $c(\boldsymbol{x}_t)$. The experimental results empirically validate our theoretical result concerning the fact that $V_T$ grows sublinearly in $T$. Furthermore, notice that the expected costs (and therefore also the expected utilities, as in zero-sum games we have that $\boldsymbol{\omega}^\star = \boldsymbol{\theta}^\star$) converge to the threshold $\gamma$. This is also reflected on the cumulative regret that turns out to be negative. The reason for such a behavior lies in the fact the strategy against which the cumulative regret is computed is over-constrained with respect to the strategy chosen at each round $t \in [T]$. Indeed, the relaxation of the cost constraint allows the expected utility to exceed the value achieved by the optimal strategy.