Exploiting Opponents Subject to Utility Constraints in Extensive-Form Games

Martino Bernasconi
Politecnico di Milano
Milan, Italy
martino.bernasconideluca@polimi.it

Federico Cacciamani
Politecnico di Milano
Milan, Italy
federico.cacciamani@polimi.it

Simone Fioravanti
GSSI
L'Aquila, Italy
simone.fioravanti@gssi.it

Nicola Gatti
Politecnico di Milano
Milan, Italy
nicola.gatti@polimi.it

Alberto Marchesi
Politecnico di Milano
Milan, Italy
alberto.marchesi@polimi.it

Francesco Trovò
Politecnico di Milano
Milan, Italy
francesco1.trovo@polimi.it

ABSTRACT

Recently, game-playing agents based on AI techniques have demonstrated super-human performance in several sequential games, such as chess, Go, and poker. Surprisingly, the multi-agent learning techniques that allowed to reach these achievements do *not* take into account the actual behavior of the human player, potentially leading to an impressive gap in performances. In this paper, we address the problem of designing artificial agents that learn how to effectively exploit unknown human opponents while playing repeatedly against them in an *online* fashion. We study the case in which the agent's strategy during each repetition of the game is subject to constraints ensuring that the human's expected utility is within some lower and upper thresholds. Our framework encompasses several real-world problems, such as human *engagement* in repeated game playing and human education by means of *serious games*. As a first result, we formalize a set of linear inequalities encoding the conditions that the agent's strategy must satisfy at each iteration in order to *not* violate the given bounds for the human's expected utility. Then, we use such formulation in an upper confidence bound algorithm, and we prove that the resulting procedure suffers from sublinear regret and guarantees that the constraints are satisfied with high probability at each iteration. Finally, we empirically evaluate the convergence of our algorithm on standard testbeds of sequential games[1].

KEYWORDS

Algorithmic Game Theory; Multiagent Learning; Online Learning; Sequential Decision Making

1 INTRODUCTION

Algorithmic game theory and machine learning have recently contributed to groundbreaking achievements in artificial intelligence, leading to the deployment of artificial agents that defeated top human professionals in several recreational games. See, for example, the well-known milestones achieved in chess [10], Go [28], and poker [8, 9]. Surprisingly, multi-agent learning techniques that have been recently employed in such settings learn how to defeat humans without taking into account their actual behavior. Indeed, they learn strategies by simulating millions of plays in a self-play approach, without including any human player in the learning process. A direct effect of this methodology is that the resulting artificial agents do *not* adapt to the actual capabilities of humans, potentially leading to an impressive gap in performances compared to the case in which the agent learns strategies while taking them into account.

In this paper, we address the problem of designing artificial agents that learn how to effectively exploit unknown human opponents while playing repeatedly against them in an *online* fashion. In particular, we study the case in which the strategy that the agent plays during each repetition of the game is subject to constraints ensuring that the human's expected utility is within some lower and upper thresholds. Our framework models several real-world human-agent interactions, and it begets additional technical challenges compared to simple pure-exploitation scenarios.

One prominent application of our framework is when the artificial agent's goal is not only exploiting the human opponent, but also ensuring that he/she remains *engaged* in the game. Guaranteeing humans' engagement is crucial when designing artificial agents that play repeatedly against humans. Indeed, when playing against a super-human agent, most human players drop out from game playing, since they realize they are losing too often against it. As Egri-Nagy and Törmänen [14] sharply observe, it is "*hopeless and frustrating to play against an AI, since it is practically impossible to win*". Different forms of engagement can be imagined (see, *e.g.*, [1] for examples in computer games). In our framework, the human's engagement is modeled with the threshold constraints over his/her expected utility, with the following rationale. If the utility value falls under a given satisfaction threshold, then the human will get bored playing, as he/she loses too much and believes he/she has no hope to win. On the other hand, if such value raises above another given threshold, then the human will get bored since he/she is winning too often.

Another application scenario for our framework is that of *serious games* [13], whose purpose is to educate humans by asking them to perform tasks engagingly. If some tasks are excessively hard for the actual human's capabilities, then the human will give up the training of the entire set of tasks, since he/she is sure that he/she will never be able to address them. On the other hand, if some tasks are excessively easy, the human will give up the training since

[1] An extended version of this paper has been published at NeurIPS 2021 [7].

Appears at the 1st Workshop on Learning with Strategic Agents (LSA 2022). Held as part of the Workshops at the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), N. Bishop, M. Han, L. Tran-Thanh, H. Xu, H. Zhang (chairs), May 9–10, 2022, Online.

he/she is sure that he/she can solve all remaining tasks. Serious games are used in many different fields, such as, *e.g.*, military [11], transportation [26], urban planning [25], and healthcare [30], and can be modeled as general-sum games between a human learner and a computer teacher [23].

Original contributions. We study two-player sequential (*i.e.*, with extensive form) games in which an artificial agent repeatedly plays against an unknown human opponent. We assume that the human has a fixed stochastic behavior and he/she does *not* learn over time. We do not make any structural assumption on the human's strategy, which requires the agent learning a probability distribution for each decision point of the human. While this is a crucial first step towards a more complex setting in which the human learns over time, the resulting model still presents several technical challenges that are worth to be investigated. First, we show how to derive, after each game repetition, a confidence region for the human's strategy such that his/her actual strategy stays within it with high probability. We show that such region is characterized by a set of linear constraints defined over the sequence-form strategy space. Notice that, during each game repetition, the agent only observes a partial sample of the human's strategy, made by the human's actions on the path in the game tree followed during play. By exploiting strong duality and the specific structure of the confidence region, we show that the thresholds constraints on the human's expected utility can be formulated as a set of linear inequalities, whose cardinality is linear in the size of the game tree. In particular, these constraints describe a subspace of the agent's sequence-form strategy space such that, for every possible human's strategy in the confidence region, the human's expected utility is within the given thresholds. We also derive a linear program with a linear number of constraints and variables to find the best agent's strategy satisfying the constraints. Then, we design an upper confidence bound algorithm, called COX-UCB, and we prove that it suffers from a sublinear regret and guarantees that the aforementioned constraints are satisfied with high probability at every iteration. Finally, we empirically evaluate the convergence of our algorithm on standard testbeds and show that our bounds are asymptotically tight. [2]

Related works. Our work is mainly related to opponent modeling, whose primary goal is to build models describing the behavior of one or multiple opponents from past interactions. Many methods are known in the literature. Specifically, Mealing and Shapiro [24] and Foerster et al. [15] propose a method to infer the parameters of the opponents' policies from the data collected during past interactions. Instead, several other works propose methods that use beliefs over a fine set of opponents' types distinguishing for their behavior: Albrecht and Ramamoorthy [3] treat types as black-box mappings, Albrecht and Stone [4] infer parameters for the types, Barrett and Stone [6] use deep learning to explicitly learn models, while He and Boyd-Graber [20] do the same implicitly. These methods usually require huge amount of training data to be precise and adaptable to new opponents. In a recent work, Wu et al. [31] propose a learning-to-exploit deep framework for implicit opponent modeling and an adversarial training procedure to automatically generate opponents so as to reduce the data needed for training. An approach that has more common ground with ours is the one adopted by Ganzfried and Sandholm [16], who propose a game-theoretic approach to develop a deviation-based best-response algorithm. In particular, the work builds an opponent model based on the deviations between the opponent's strategy and a precomputed approximate equilibrium, and, then, computes a best response in real-time. The only works providing theoretical guarantees are [17, 18], which deal with safe opponent exploitation, *i.e.*, guaranteeing a certain agent's payoff in expectation, given any opponent's strategy. Differently, in our setting, the goal is to guarantee a certain human's expected utility.

[2] All the omitted proofs can be found in the Appendix.

2 PRELIMINARIES

In this section, we review the basic concepts and definitions related to sequential games that we need in the rest of this work (see the book by Shoham and Leyton-Brown [27] for more details).

Extensive-form games. We focus on *two-player extensive-form games* (EFGs) with imperfect information in which an artificial agent faces a human opponent. We denote by i the agent player, while j is the human. Then, the set of players is $P \cup \{c\}$, where we let $P \coloneqq \{i, j\}$ and c denotes a *chance player* that selects actions according to fixed known probabilities, representing exogenous stochasticity. An EFG is usually defined by means of a *game tree*, where H is the set of nodes of the tree and $Z \subseteq H$ is the subset of terminal nodes, which are the leaves of the game tree. A node $h \in H$ is identified by the ordered list of actions encountered on the path from the root of the game tree to the node. Given a non-terminal node $h \in H \setminus Z$, we let $P(h) \in P \cup \{c\}$ be the unique player who acts at h and $A(h)$ be the set of actions he/she has available. We let $u_i, u_j : Z \to \mathbb{R}$ be the payoff functions of players i and j, respectively. Moreover, we denote by $p_c : H \to [0, 1]$ the function assigning each node $h \in H$ to the product of probabilities of chance moves on the path from the root of the game tree to h. Imperfect information is encoded by using *information sets* (infosets for short). A player i's infoset I groups nodes belonging to player i that are indistinguishable for him/her, that is, $I \subseteq H \setminus Z$ is such that $P(h) = P(k) = i$ and $A(h) = A(k)$ for any pair of nodes $h, k \in I$. We let $\mathcal{I}$ be the set of player i's infosets, which define a partition of the set of player i's non-terminal nodes $\{h \in H \setminus Z \mid P(h) = i\}$. Moreover, with a slight abuse of notation, we let $A(I)$ be the set of actions available at all the nodes in infoset $I \in \mathcal{I}$. Analogously, we define $\mathcal{J}$ as the set of player j's infosets, while, for any infoset $J \in \mathcal{J}$, we let $A(J)$ be the set of actions available at nodes in J. We focus on games with *perfect recall* in which infosets are such that no player forgets information once acquired.

The sequence form of EFGs. The *sequence form* is a compact way of representing EFGs with perfect recall [21, 29], where the pure strategies of a player—specifying an action at each infoset of that player—are replaced by the concept of sequence. Any node $h \in H$ defines a *sequence* $\sigma_i(h)$ of player i, which is identified by the ordered list of player i's actions on the path from the root of the game tree to h. In perfect-recall EFGs, all the nodes belonging to an infoset $I \in \mathcal{I}$ of player i define the same player i's sequence, which, by overloading notation, we denote by $\sigma_i(I)$. Sequence $\sigma_i(I)$ can be extended

by appending any action $a \in A(I)$ available at I at its end, obtaining another valid player i's sequence that we denote as $\sigma_i(I)a$. Then, the set of player i's sequences is $\Sigma_i := \{\sigma_i(I)a \mid I \in \mathcal{I}, a \in A(I)\} \cup \{\varnothing\}$, where $\varnothing$ is the empty sequence (defined by all the nodes such that player i never plays before them in the game tree). Analogously, we define $\Sigma_j := \{\sigma_j(J)a \mid J \in \mathcal{J}, a \in A(J)\} \cup \{\varnothing\}$ as the set of all player j's sequences. Mixed strategies in the sequence form are specified by defining the realization probability of each sequence. A *sequence-form strategy* of player i is denoted by a vector $\boldsymbol{x} \in [0,1]^{|\Sigma_i|}$, with $\boldsymbol{x}[\sigma_i]$ being the realization probability of sequence $\sigma_i \in \Sigma_i$.[3] To be well defined, a sequence-form strategy must satisfy a set of linear constraints, ensuring that realization probabilities of sequences encode a valid probability distribution over actions at each infoset. Formally, any $\boldsymbol{x} \in [0,1]^{|\Sigma_i|}$ must satisfy:

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

Constraints (1) can be written as $\boldsymbol{F}_i\boldsymbol{x} = \boldsymbol{f}_i$, where $\boldsymbol{F}_i \in \{-1,0,1\}^{\ell}$, where $\ell := (|\mathcal{I}| + 1) \times |\Sigma_i|$, and $\boldsymbol{f}_i \in \{0,1\}^{|\mathcal{I}|+1}$ are a suitably-defined matrix and vector, respectively (see [29] for their definitions). Analogously, we denote sequence-form strategies of player j as vectors $\boldsymbol{y} \in [0,1]^{|\Sigma_j|}$ that satisfy the condition $\boldsymbol{F}_j\boldsymbol{y} = \boldsymbol{f}_j$, which is defined by linear constraints analogous to Constraints (1). For the ease of presentation, we let $\mathcal{X} := \left\{\boldsymbol{x} \in [0,1]^{|\Sigma_i|} \mid \boldsymbol{F}_i\boldsymbol{x} = \boldsymbol{f}_i\right\}$ and $\mathcal{Y} := \left\{\boldsymbol{y} \in [0,1]^{|\Sigma_j|} \mid \boldsymbol{F}_j\boldsymbol{y} = \boldsymbol{f}_j\right\}$ be the sets of all sequence-form strategies of players i and j, respectively. Finally, given two strategies $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \mathcal{Y}$, it is easy to check that player i's expected payoff can be written as the bilinear form $\boldsymbol{x}^\top \boldsymbol{U}_i \boldsymbol{y}$, where $\boldsymbol{U}_i \in \mathbb{R}^{|\Sigma_i| \times |\Sigma_j|}$ is player i's sequence-form *utility matrix*, defined $\forall \sigma_i \in \Sigma_i, \forall \sigma_j \in \Sigma_j$ as:

$$\boldsymbol{U}_i[\sigma_i, \sigma_j] := \sum_{z \in Z : \sigma_i(z) = \sigma_i \wedge \sigma_j(z) = \sigma_j} p_c(z) u_i(z).$$

Analogously, the sequence-form utility matrix of player j is $\boldsymbol{U}_j \in \mathbb{R}^{|\Sigma_i| \times |\Sigma_j|}$, and, thus, $\boldsymbol{x}^\top \boldsymbol{U}_j \boldsymbol{y}$ is his/her expected payoff given two sequence-form strategies $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \mathcal{Y}$.

Additional notation. Given two sequences $\sigma_i, \sigma_i' \in \Sigma_i$ of player i, we write $\sigma_i \sqsubseteq \sigma_i'$ to denote that σ_i is a *sub-sequence* of σ_i'; formally, this is the case whenever the ordered list of actions identified by σ_i is a prefix of that of σ_i'. Similarly, we use notation $\sigma_j \sqsubseteq \sigma_j'$ for two sequences $\sigma_j, \sigma_j' \in \Sigma_j$ of player j. Moreover, given a player i's sequence-form strategy $\boldsymbol{x} \in \mathcal{X}$ and a player j's infoset $J \in \mathcal{J}$, we let $\rho_{-j}(J, \boldsymbol{x})$ be the probability of reaching J given that player j plays so as to reach it and player i plays $\boldsymbol{x}$ (also accounting for chance probabilities); formally $\rho_{-j}(J, \boldsymbol{x}) := \sum_{h \in J} \boldsymbol{x}[\sigma_i(h)] p_c(h)$.

3 EXPLOITING OPPONENTS UNDER UTILITY CONSTRAINTS

We study settings in which the agent player i repeatedly faces the human opponent j in a two-player EFG. Our goal is the design of agents that learn the strategy of the human so as to effectively exploit it, while at the same time guaranteeing that the human's expected utility remains under control during the entire repeated interaction.[4] In the rest of this section, we formally introduce our problem and provide a general overview of the approach we undertake to tackle it.

We let T be the number of times the EFG is played. Player j plays according to the same sequence-form strategy $\boldsymbol{y}^* \in \mathcal{Y}$ at each iteration $t \in [T]$.[5] This strategy is unknown to player i. On the other hand, at iteration t, player i selects and plays a strategy $\boldsymbol{x}^t \in \mathcal{X}$. Then, at the end of the iteration, player i receives as feedback a sequence of player j's actions $\sigma_j^t \in \Sigma_j$, which is defined by the path in the game tree followed during game playing at that iteration (notice that σ_j^t is made of actions sampled according to player j's strategy $\boldsymbol{y}^*$). In the following, for $t \in [T]$, we let $\mathcal{H}^t$ be the history of feedbacks received by the agent player i up to iteration t (included), namely $\mathcal{H}^t := \left(\sigma_j^1, \sigma_j^2, \dots \sigma_j^t\right)$.

To ensure that player j's utility is kept under control during the repeated interaction, player i must play strategies $\boldsymbol{x}^t$ such that the resulting player j's expected utilities are within some given thresholds. The lower limit of the range ensures that the agent does *not* over-exploit the human. On the other hand, the upper limit of the range guarantees that the expected payoff of the human is *not* too high. Formally, we require $\boldsymbol{x}^t \in \mathcal{X}^t$, where the subset $\mathcal{X}^t \subseteq \mathcal{X}$ is defined as follows:

Definition 3.1 (Utility-constrained Strategy Set). Let $t \in [T]$ and $\delta \in (0,1)$. Given a lower limit $\alpha \in \mathbb{R}$ and an upper limit $\beta \in \mathbb{R}$, we define the *utility-constrained strategy set* $\mathcal{X}^t \subseteq \mathcal{X}$ at iteration t as the set of player i's sequence-form strategies $\boldsymbol{x} \in \mathcal{X}$ such that $\mathbb{P}\left(\alpha \leq \boldsymbol{x}^\top \boldsymbol{U}_j \boldsymbol{y}^* \leq \beta\right) \geq 1 - \delta$, with respect to the randomness of the history $\mathcal{H}^{t-1}$ of feedbacks observed by player i up to iteration $t-1$ (included).[6]

After T game repetitions, given the strategies $\boldsymbol{x}^t \in \mathcal{X}^t$ played by the agent player i during iterations $t \in [T]$, we measure his/her performance by means of the following notion of regret:

$$R^T := \sum_{t=1}^{T} \left[\max_{\boldsymbol{x}^* \in \mathcal{X}^t} \left(\boldsymbol{x}^*\right)^\top \boldsymbol{U}_i \boldsymbol{y}^* - \left(\boldsymbol{x}^t\right)^\top \boldsymbol{U}_i \boldsymbol{y}^* \right],$$

which represents how much player i would have gained in expectation by playing a utility-maximizing strategy in the utility-constrained strategy set $\mathcal{X}^t$ rather than $\boldsymbol{x}^t$, at each iteration $t \in [T]$. The goal that we pursue in the rest of this work is to achieve sublinear regret, that is $R^T = o(T)$, while at the same time guaranteeing that the played strategies $\boldsymbol{x}^t$ satisfy the constraints defined by the sets $\mathcal{X}^t$.

Overview of our results. In what follows, we give a brief sketch of the approach we adopt to tackle the problem. We propose a learning algorithm for the agent player i, which we call *Constrained Opponent eXploitation with Upper Confidence Bounds* (COX-UCB).

[3] In this work, we denote vectors by bold symbols. Given a finite set S of dimension $|S| = d$, we denote by $\boldsymbol{v} \in \mathbb{R}^{|S|}$ a d-dimensional vector indexed over S, with $\boldsymbol{v}[s]$ being its component corresponding to $s \in S$.

[4] The methodology that we propose in this paper can also be adapted to control the agent's expected utility, rather than the one of the human player.

[5] In this work, given $n \in \mathbb{N}_+$ we denote by $[n]$ the set $\{1, \dots, n\}$ of the first n natural numbers.

[6] In the rest of this work, we make implicit the dependency of $\mathcal{X}^t$ from δ, α, and β, as the values of these parameters will be clear from context.

It builds on two core components. The first one deals with the construction of the utility-constrained strategy set $\mathcal{X}^t$ at each iteration $t \in [T]$. It works by building a confidence region $\mathcal{Y}^{t-1} \subseteq \mathcal{Y}$ for player j's strategy, using the history $\mathcal{H}^{t-1}$ of feedbacks observed up the previous iteration $t-1$. This is such that the true (unknown) strategy $\boldsymbol{y}^*$ lies within $\mathcal{Y}^{t-1}$ with probability at least $1-\delta$, for some fixed confidence level $\delta \in (0,1)$. Then, the utility-constrained strategy set $\mathcal{X}^t$ can be characterized by a set of linear inequalities that exploits the structure of the confidence region $\mathcal{Y}^{t-1}$. A detailed formal treatment of this first component is provided in Section 4. The second core component consists in a rule to select the strategy $\boldsymbol{x}^t \in \mathcal{X}^t$ to play at each iteration $t \in [T]$. We propose an approach based on the *optimism in face of uncertainty* principle. More details on this second component can be found in Section 5, together with the regret bounds attained by the algorithm. We refer the reader to Algorithm 1 for a general sketch of our COX-UCB algorithm, where two alternative implementations of the procedure SelectStrategy$(\mathcal{X}^t, \mathcal{Y}^{t-1})$ will be given in Algorithms 2 and 3 in Section 5.

Algorithm 1 COX-UCB

$t \leftarrow 1$
while $t \leq T$ **do**
  Build confidence region $\mathcal{Y}^{t-1}$ from history of past feedbacks $\mathcal{H}^{t-1}$
  Use $\mathcal{Y}^{t-1}$ to build the utility-constrained strategy set $\mathcal{X}^t$
  $\boldsymbol{x}^t \leftarrow$ SelectStrategy$(\mathcal{X}^t, \mathcal{Y}^{t-1})$
  Play the game according to strategy $\boldsymbol{x}^t$
  Observe player j's sequence σ_j^t, obtained from the path in the game tree followed during play
  $\mathcal{H}^t \leftarrow \mathcal{H}^{t-1} \cup \{\sigma_j^t\}$
  $t \leftarrow t+1$

4 HOW TO CONTROL THE HUMAN'S EXPECTED UTILITY

In this section, we provide the formal details on the construction of the utility-constrained strategy set $\mathcal{X}^t$ at each iteration $t \in [T]$ (Definition 3.1). We split the section into two main parts:

- Subsection 4.1 shows how to use the history $\mathcal{H}^t$ of feedbacks observed by player i up to iteration t to derive a confidence region $\mathcal{Y}^t \subseteq \mathcal{Y}$ for player j's strategy $\boldsymbol{y}^*$, such that $\boldsymbol{y}^*$ lies within $\mathcal{Y}^t$ with probability at least $1-\delta$ for some fixed confidence $\delta \in (0,1)$;
- Subsection 4.2 describes how to exploit the confidence region $\mathcal{Y}^{t-1}$ built using feedbacks observed up to iteration $t-1$ to construct a set of linear constraints that fully characterize sequence-form strategies in the utility-constrained strategy set $\mathcal{X}^t$ at iteration t.

4.1 Building a confidence region for the human's strategy

Let us recall that, at each iteration $t \in [T]$, player i observes a player j's sequence σ_j^t determined by selecting actions to play during the game according to the sequence-form strategy $\boldsymbol{y}^*$. We build the desired high-probability confidence region $\mathcal{Y}^t$ by exploiting information provided by observed sequences to derive, for each player j's infoset $J \in \mathcal{J}$, appropriate confidence intervals for the realization probabilities $\boldsymbol{y}^*[\sigma_j(J)a]$ of sequences $\sigma_j(J)a$ terminating with an action $a \in A(J)$ at J.[7]

The case of a single infoset. Before showing our general technique, it is useful to present the easier setting in which player j has a unique infoset, and, thus, his/her strategy is defined as a probability distribution $\boldsymbol{p} \in \Delta^{|A|}$, where, with an abuse of notation, A denotes the finite set of actions available at the infoset. In this case, player i observes t actions $a^1, \dots, a^t \in A$ sampled independently according to $\boldsymbol{p}$. Then, a natural estimator for $\boldsymbol{p}$ is the empirical frequency of actions $\boldsymbol{p}^t \in \Delta^{|A|}$, defined so that $\boldsymbol{p}^t[a] \coloneqq \frac{1}{t}\sum_{\tau=1}^{t} \mathbb{1}\{a^\tau = a\}$ for every $a \in A$. By noticing that $t\,\boldsymbol{p}^t$ is a random variable following a multinomial distribution with parameters t and $\boldsymbol{p}$, that is $t\,\boldsymbol{p}^t \sim \mathcal{M}(t; \boldsymbol{p})$, the following lemma by Devroye [12] can be used to derive the desired confidence intervals for the probabilities $\boldsymbol{p}[a]$.[8]

Lemma 4.1 (Lemma 3 by Devroye [12]). *Let $\boldsymbol{p} \in \Delta^{|A|}$ and $a^1, \dots, a^t \in A$ be t actions sampled independently according to $\boldsymbol{p}$. Then, for any $0 < \delta \leq 3\exp(-4|A|/5)$, it holds:*

$$\mathbb{P}\left(\sum_{a \in A}\left|\boldsymbol{p}^t[a] - \boldsymbol{p}[a]\right| \leq 5\sqrt{\frac{\ln(3/\delta)}{t}}\right) \geq 1-\delta.$$

By exploiting the fact that $\boldsymbol{p} \in \Delta^{|A|}$, we can refine the result in Lemma 4.1 by giving bounds that hold for each component of $\boldsymbol{p}$ separately (see Lemma 4.2 below). This additional step is crucial when building confidence intervals for realization probabilities $\boldsymbol{y}^*[\sigma_j(J)a]$ at infoset J in general.

Lemma 4.2. *Let $\boldsymbol{p} \in \Delta^{|A|}$ and $a^1, \dots, a^t \in A$ be t actions sampled independently according to $\boldsymbol{p}$. Then, for any $0 < \delta \leq 3\exp(-4|A|/5)$, it holds:*

$$\mathbb{P}\left(\bigcap_{a \in A}\left\{\left|\boldsymbol{p}^t[a] - \boldsymbol{p}[a]\right| \leq \frac{5}{2}\sqrt{\frac{\ln(3/\delta)}{t}}\right\}\right) \geq 1-\delta.$$

Next, we generalize the approach described above to the general case of any infosets structure.

First, we introduce some useful random variables. For every iteration $t \in [T]$, player j's infoset $J \in \mathcal{J}$, and action $a \in A(J)$, we let $O^t(J,a) \coloneqq \mathbb{I}\left\{\sigma_j(J)a \sqsubseteq \sigma_j^t\right\}$ be a random variable that is equal to 1 if and only if player j played action a at infoset J during iteration t, while it is equal to 0 otherwise. It is easy to check that $O^t(J,a)$ follows a Bernoulli distribution with parameter $p^t(J,a) \coloneqq \boldsymbol{y}^*[\sigma_j(J)a]\,\rho_{-j}^t(J)$, where, for the ease of presentation, we let $\rho_{-j}^t(J) \coloneqq \rho_{-j}(J, \boldsymbol{x}^t)$ be the contribution to the probability of

[7] Let us remark that deriving $\mathcal{Y}^t$ is made considerably challenging by the fact that, at each $t \in [T]$, only the sequence σ_j^t of actions actually played by player j is observed. On the other hand, if player i would be able to observe the actual pure strategy selected by player j at t, the problem would admit a much easier solution consisting in building a single confidence interval for player j's average strategy.

[8] In this work, we denote by $\Delta^{|S|}$ the $(|S|-1)$-dimensional simplex indexed over the finite set S. Moreover, we denote by $\mathbb{1}\{\cdot\}$ the indicator function for the event enclosed in curly braces, while $\mathcal{M}(n; \boldsymbol{v})$ denotes a multinomial probability distribution, where $n \in \mathbb{N}_+$ is the number of trials and $\boldsymbol{v} \in \Delta^{|S|}$ is a vector defining the probabilities of observing each element in the finite set S.

reaching infoset J due to player i's strategy $\boldsymbol{x}^t$ and chance probabilities.[9] The random variables $O^t(J,a)$ are instrumental for defining $N^t(J,a) \coloneqq \sum_{\tau=1}^{t} O^\tau(J,a)$, which represents the number of times a is played at J up to iteration t. Intuitively, variables $N^t(J,a)$ of infoset $J \in \mathcal{J}$ play the same role as the random vector $t\,\boldsymbol{p}^t$ in the single-infoset case.

We follow an approach analogous to that of the single-infoset case at each player j's infoset, and, then, put all the resulting confidence intervals together to define $\mathcal{Y}^t$. To do so, we need to circumvent the following issue: at each infoset $J \in \mathcal{J}$, the random variables $N^t(J,a)$ for $a \in A(J)$ are *not* jointly distributed as a multinomial, preventing a direct application of Lemma 4.1. We deal with this by adding a fictitious action at each infoset, so that random variables $N^t(J,a)$ are multinomially distributed for each $J \in \mathcal{J}$. Then, the fictitious action can be easily factored out by using Lemma 4.2.

Let $A_\diamond(J) \coloneqq A(J) \cup \{a_\diamond\}$ be the new action set at $J \in \mathcal{J}$, with $a_\diamond$ denoting the fictitious action. For $t \in [T]$, we define $O^t(J,a_\diamond) \coloneqq \mathbb{1}\left\{\sigma_j(J)a \not\sqsubseteq \sigma_j^t\right\}$ as a random variable equal to 1 if and only if it is *not* the case that a is played at J during iteration t (this also includes all the cases in which J is not reached), and 0 otherwise. Clearly, $O^t(J,a_\diamond)$ follows a Bernoulli distribution with parameter $p^t(J,a_\diamond) \coloneqq 1 - \sum_{a\in A(J)} p^t(J,a)$. Moreover, we let $N^t(J,a_\diamond) \coloneqq \sum_{\tau=1}^{t} O^\tau(J,a_\diamond)$. Then, for each player j's infoset $J \in \mathcal{J}$, we define $\boldsymbol{n}_J^t \in \mathbb{N}^{|A_\diamond(J)|}$ as a random vector such that $\boldsymbol{n}_J^t[a] \coloneqq N^t(J,a)$ for every $a \in A_\diamond(J)$. By letting $\bar{\rho}_{-j}^t(J) \coloneqq \frac{1}{t}\sum_{\tau=1}^{t} \rho_{-j}^\tau(J)$, we have:

$$\begin{aligned}\mathbb{E}\left[N^t(J,a)\right] &= \sum_{\tau=1}^{t} \mathbb{E}\left[O^\tau(J,a)\right] \\ &= \boldsymbol{y}^*[\sigma_j(J)a] \sum_{\tau=1}^{t} \rho_{-j}^\tau(J) \\ &= t\,\bar{\rho}_{-j}^t(J)\boldsymbol{y}^*[\sigma_j(J)a],\end{aligned}$$

while it is easy to check that

$$\mathbb{E}\left[N^t(J,a_\diamond)\right] = t - t\,\bar{\rho}_{-j}^t(J) \sum_{a\in A(J)} \boldsymbol{y}^*[\sigma_j(J)a].$$

As a result, we conclude that $\boldsymbol{n}_J^t$ follows a multinomial distribution, circumventing our initial issue. Formally:

$$\boldsymbol{n}_J^t \sim \mathcal{M}(t,\boldsymbol{v}), \boldsymbol{v} \in \Delta^{|A_\diamond(J)|} \text{ s.t.}$$

$$\boldsymbol{v}[a] = \begin{cases} \bar{\rho}_{-j}^t(J)\boldsymbol{y}^*[\sigma_j(J)a] & \text{if } a \in A(J) \\ 1 - \bar{\rho}_{-j}^t(J)\sum_{a\in A(J)} \boldsymbol{y}^*[\sigma_j(J)a] & \text{if } a = a_\diamond \end{cases}.$$

By exploiting this last observation and using Lemmas 4.1 and 4.2, we provide confidence intervals defined locally at each infoset $J \in \mathcal{J}$ for the realization probabilities $\boldsymbol{y}^*[\sigma_j(J)a]$ of sequences $\sigma_j(J)a$ terminating with an action $a \in A(J)$ at J. By letting $\boldsymbol{y}^t \in \mathbb{R}^{|\Sigma_j|}$ be such that:

$$\boldsymbol{y}^t[\sigma_j(J)a] \coloneqq \frac{N^t(J,a)}{t\bar{\rho}_{-j}^t(J)} \quad \forall J \in \mathcal{J}, \forall a \in A(J), \tag{2}$$

we have the following lemma:

[9] In the rest of this work, we assume w.l.o.g. that $\rho_{-j}^t(J) > 0$ for any t and J. This is possible thanks to the fact that strategies $\boldsymbol{x}^t$ selected by the algorithm proposed in Section 5 always ensure that such conditions hold.

Lemma 4.3. *Let $J \in \mathcal{J}$ be a player j's infoset. Then, for any $0 < \delta \leq 3\exp(-4|A(J)|/5)$ it holds:*

$$\mathbb{P}\left(\bigcap_{a\in A(J)} \left\{\left|\boldsymbol{y}^t[\sigma_j(J)a] - \boldsymbol{y}^*[\sigma_j(J)a]\right| \leq \frac{5}{2\bar{\rho}_{-j}^t(J)}\sqrt{\frac{\ln(3/\delta)}{t}}\right\}\right) \geq 1-\delta.$$

Lemma 4.3 shows that one can use $\boldsymbol{y}^t[\sigma_j(J)a]$ to estimate the realization probability $\boldsymbol{y}^*[\sigma_j(J)a]$ of some sequence $\sigma_j(J)a$, where $\boldsymbol{y}^t[\sigma_j(J)a]$ represents the observed frequency of action a at infoset J, adjusted by the average probability of reaching J due to other players, namely $\bar{\rho}^t(J)$. Thus, $\boldsymbol{y}^t$ generalizes the empirical frequency $\boldsymbol{p}^t$ in the single-infoset case to the sequence-form setting. Moreover, Lemma 4.3 gives high-probability confidence intervals for probabilities $\boldsymbol{y}^*[\sigma_j(J)a]$ at every infoset $J \in \mathcal{J}$, which is used in the following theorem that gives us the desired high-probability confidence region $\mathcal{Y}^t$.

Theorem 4.4. *For every player j's infoset $J \in \mathcal{J}$, let $\delta_J \in (0,1)$ be such that the condition in Lemma 4.3 is satisfied and $\sum_{J\in\mathcal{J}} \delta_J < 1$. Then, $\mathbb{P}\left(\boldsymbol{y}^* \in \mathcal{Y}^t\right) \geq 1-\delta$, where $\delta \coloneqq \sum_{J\in\mathcal{J}} \delta_J$ and*

$$\mathcal{Y}^t \coloneqq \left\{\boldsymbol{y} \in \mathcal{Y} : \left|\boldsymbol{y}^t[\sigma_j(J)a] - \boldsymbol{y}^*[\sigma_j(J)a]\right| \leq \frac{5}{2\bar{\rho}_{-j}^t(J)}\sqrt{\frac{\ln(3/\delta_J)}{t}} \quad \forall J \in \mathcal{J}, \forall a \in A(J)\right\}.$$

4.2 Constructing the utility-constrained strategy set

We show how to construct the utility-constrained strategy set $\mathcal{X}^t$—for some $\delta \in (0,1)$ and $\alpha, \beta \in \mathbb{R}$—by exploiting the high-probability confidence region $\mathcal{Y}^{t-1}$ (see Theorem 4.4). Our approach is to force the condition that each $\boldsymbol{x}$ in the utility-constrained strategy set must satisfy with high probability (see Definition 3.1 for such condition) for every $\boldsymbol{y} \in \mathcal{Y}^{t-1}$; formally, we require $\alpha \leq \boldsymbol{x}^\top \boldsymbol{U}_j \boldsymbol{y} \leq \beta$ for all $\boldsymbol{y} \in \mathcal{Y}^t$. By definition of $\mathcal{Y}^{t-1}$, this guarantees that such constraint holds also for $\boldsymbol{y}^*$ with probability at least $1-\delta$. Formally, we define

$$\mathcal{X}^t \coloneqq \left\{\boldsymbol{x} \in \mathcal{X} : \max_{\boldsymbol{y}\in\mathcal{Y}^{t-1}} \boldsymbol{x}^\top \boldsymbol{U}_j \boldsymbol{y} \leq \beta \;\wedge\; \min_{\boldsymbol{y}\in\mathcal{Y}^{t-1}} \boldsymbol{x}^\top \boldsymbol{U}_j \boldsymbol{y} \geq \alpha\right\},$$

so that, for any $\boldsymbol{x} \in \mathcal{X}^t$, the upper limit β and the lower limit α are satisfied for every $\boldsymbol{y} \in \mathcal{Y}^{t-1}$.

In what follows, we characterize the set $\mathcal{X}^t$ by means of a set of linear inequalities. We formulate the constrained maximization problem, *i.e.*, $\max_{\boldsymbol{y}\in\mathcal{Y}^{t-1}} \boldsymbol{x}^\top \boldsymbol{U}_j \boldsymbol{y}$, using the following linear program:

$$\max_{\boldsymbol{y}\geq\boldsymbol{0}} \boldsymbol{x}^\top \boldsymbol{U}_j \boldsymbol{y} \quad \text{s.t.} \tag{3a}$$

$$F_j \boldsymbol{y} = \boldsymbol{f}_j \tag{3b}$$

$$\boldsymbol{y} \leq \boldsymbol{y}^{t-1} + \boldsymbol{\epsilon}^{t-1} \tag{3c}$$

$$\boldsymbol{y} \geq \boldsymbol{y}^{t-1} - \boldsymbol{\epsilon}^{t-1}, \tag{3d}$$

where we define $\boldsymbol{\epsilon}^{t-1}[\sigma_j(J)a] \coloneqq \frac{5}{2\bar{\rho}^t_{-j}(J)}\sqrt{\frac{\ln(3/\delta_J)}{t}}$ for $J \in \mathcal{J}$ and $a \in A(J)$, for some $\delta_J \in (0,1)$ that satisfy the condition in Lemma 4.3 and $\sum_{J\in\mathcal{J}} \delta_J \leq \delta$. Notice that, by recalling the definition of $\boldsymbol{y}^{t-1}$ in Equation (2) and that of $\mathcal{Y}^{t-1}$ in Theorem 4.4, Constraints (3b), (3c), and (3d) correctly encode the fact that $\boldsymbol{y}$ must belong to the set $\mathcal{Y}^{t-1}$.

The dual of Problem (3) reads as: $\min_{\boldsymbol{u}\geq 0} \boldsymbol{b}^\top \boldsymbol{u}$ s.t. $\boldsymbol{A}^\top \boldsymbol{u} \geq \boldsymbol{U}_j^\top \boldsymbol{x}$, where $\boldsymbol{u} \in \mathbb{R}^{(|\mathcal{J}|+1)\times 2|\Sigma_j|}$ is a vector of dual variables, while $\boldsymbol{b}$ and $\boldsymbol{A}$ are suitably-defined vector and matrix, respectively. By strong duality, the optimal dual objective equates the optimal primal objective, and, thus, given any player i's strategy $\boldsymbol{x} \in \mathcal{X}$, the condition $\max_{\boldsymbol{y}\in\mathcal{Y}^{t-1}} \boldsymbol{x}^\top \boldsymbol{U}_j \boldsymbol{y} \leq \beta$ holds if there exists a dual feasible solution $\boldsymbol{u}$ that satisfies the additional constraint that $\boldsymbol{b}^\top \boldsymbol{u} \leq \beta$.

By following an analogous reasoning for $\min_{\boldsymbol{y}\in\mathcal{Y}^{t-1}} \boldsymbol{x}^\top \boldsymbol{U}_j \boldsymbol{y}$, and letting $\boldsymbol{\omega} \in \mathbb{R}^{(|\mathcal{J}|+1)\times 2|\Sigma_j|}$ be a vector of variables of the dual problem corresponding to its linear programming formulation (similar to Problem (3)), we state the following main result:

Theorem 4.5. *Let $t \in [T]$ and $\delta \in (0,1)$. Given $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$, it holds:*

$$\mathcal{X}^t = \mathcal{X} \cap \big\{(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\omega}) : \boldsymbol{u}, \boldsymbol{\omega} \geq \boldsymbol{0},\ \boldsymbol{b}^\top \boldsymbol{u} \leq \beta,\ \boldsymbol{A}^\top \boldsymbol{u} \geq \boldsymbol{U}_j^\top \boldsymbol{x},\ -\boldsymbol{b}^\top \boldsymbol{\omega} \geq \alpha,\ -\boldsymbol{A}^\top \boldsymbol{\omega} \leq \boldsymbol{U}_j^\top \boldsymbol{x}\big\}.$$

From Theorem 4.5, it follows that $\mathcal{X}^t$ can be characterized by a polynomially-sized set of linear constraints, which achieves our initial goal.

5 HOW TO SELECT THE STRATEGY TO PLAY

In this section, we provide the implementation of the procedure SelectStrategy($\mathcal{X}^t, \mathcal{Y}^{t-1}$) in Algorithm 1. To guarantee that COX-UCB attains sub-linear regret after T iterations, *i.e.*, $R^T = o(T)$ (with high probability), we adopt an approach inspired from arm-selection strategies used in *linear* multi-armed bandit problems [5]. In particular, we propose one that uses upper confidence bounds, which is inspired by the LinUCB algorithm [2].

As a first step, we need to ensure that the algorithm performs enough exploration during game playing. This is crucial to lower bound the probabilities $\bar{\rho}^t_{-j}(J)$ that appear in the bounds defining the set $\mathcal{Y}^t$ (see Theorem 4.4), and, ultimately, to obtain sub-linear regret. In particular, at each $t \in [T]$, the COX-UCB algorithm selects a strategy $\boldsymbol{x}^t$ that belongs to the following subset of $\mathcal{X}^t$:

$$\tilde{\mathcal{X}}^t \coloneqq \big\{\boldsymbol{x} \in \mathcal{X}^t : \boldsymbol{x}[\sigma_i(z)] \geq \alpha^t \ \ \forall z \in Z\big\},$$

where the α^ts are suitably-defined parameters that decrease with the iteration number t.[10]

Algorithm 2 Strategy selection of COX-UCB

function SelStrat-UCB($\mathcal{X}^t, \mathcal{Y}^{t-1}$)
  $\boldsymbol{x}^t \leftarrow \operatorname{argmax}_{\boldsymbol{x}\in\tilde{\mathcal{X}}^t} \max_{\boldsymbol{y}\in\mathcal{Y}^{t-1}} \boldsymbol{x}^\top \boldsymbol{U}_i \boldsymbol{y}$
  return $\boldsymbol{x}^t$

[10] Notice that a naïve way of adding exploration to the algorithm would be to use an ϵ-greedy policy that plays the selected strategy $\boldsymbol{x}^t \in \mathcal{X}^t$ with probability $1-\epsilon$ and a random one with probability ϵ. However, this approach does *not* work in our setting, as it would result in a violation of the constraints on the human's expected utility. On the other hand, picking $\boldsymbol{x} \in \tilde{\mathcal{X}}^t$ assures that such constraints are satisfied.

Algorithm 3 Strategy selection of ψ-COX-UCB

function SelStrat-ψ-UCB($\mathcal{X}^t, \mathcal{Y}^{t-1}$)
  with probability $1-\psi$ do:
    $\boldsymbol{x}^t \leftarrow \operatorname{argmax}_{\boldsymbol{x}\in\tilde{\mathcal{X}}^t} \max_{\boldsymbol{y}\in\mathcal{Y}^{t-1}} \boldsymbol{x}^\top \boldsymbol{U}_i \boldsymbol{y}$
  with probability ψ do:
    $\boldsymbol{x}^t \leftarrow \operatorname{argmax}_{\boldsymbol{x}\in\tilde{\mathcal{X}}^t} \boldsymbol{x}^\top \boldsymbol{U}_i \boldsymbol{y}^{t-1}$
  return $\boldsymbol{x}^t$

The strategy selection mechanism implemented by COX-UCB is provided in Algorithm 2. It is based on the *optimism in face of uncertainty* principle, and, thus, it selects a strategy $\boldsymbol{x}^t \in \tilde{\mathcal{X}}^t$ that maximizes player i's expected payoff $\boldsymbol{x}^\top \boldsymbol{U}_i \boldsymbol{y}$ under the assumption that, for every $\boldsymbol{x} \in \tilde{\mathcal{X}}^t$, strategy $\boldsymbol{y}$ is an optimistic estimate of $\boldsymbol{y}^*$ taken from the confidence region $\mathcal{Y}^{t-1}$, that is, $\boldsymbol{y} \in \mathcal{Y}^{t-1}$ maximizes the same player i's expected payoff $\boldsymbol{x}^\top \boldsymbol{U}_i \boldsymbol{y}$. The following theorem provides a high-probability sub-linear regret guarantee for COX-UCB.

Theorem 5.1. *Let $\alpha^t \coloneqq \eta \frac{2\ln^2 t + \ln t + 1}{\sqrt{\ln t}\,(\ln t+1)^2}$ for every $t \in [T]$, where $\eta \in (0,1)$, and let $\delta \in (0,1)$. The COX-UCB algorithm attains the following regret bound with probability at least $1-\delta$:*

$$R^T \leq \frac{5}{2\eta} K_{\boldsymbol{U}_i} C \left(1 + 2\sqrt{T \ln T}\right),$$

where $K_{\boldsymbol{U}_i} \coloneqq ||\boldsymbol{U}_i||_\infty$ and C is a suitably-defined constant.

Let us remark that to obtain sub-linear regret only the term $1/\sqrt{\log t}$ is required in the definition of α^t; the other terms are added so as to obtain an analytical formula for the regret. Moreover, notice that COX-UCB needs to solve a linearly-constrained *bilinear* optimization problem at each iteration, which can be done efficiently by cutting-hedge solvers (see Section 6).

6 EXPERIMENTAL EVALUATION

We empirically evaluate the convergence of our COX-UCB algorithm on a standard testbed of Kuhn poker [22]. We report here the evaluation rank 7 Kuhn poker (*kuhn_2*).

6.1 Approximated version of the COX-UCB algorithm

The COX-UCB algorithm presented in Section 5 (Algorithm 2) solves a bilinear optimization problem at every iteration, with the optimization done over the whole utility-constrained strategy set. Furthermore, let us recall that such set is directly built from the high-confidence region for the human's strategy, which is specified at time t by an estimate $\boldsymbol{y}^t$ and bounds $\boldsymbol{\epsilon}^t$. Empirical evidence from early experiments showed that, despite having the same worst-case convergence rate, in practice the estimate $\boldsymbol{y}^t$ converges much faster to $\boldsymbol{y}^*$ than the bound $\boldsymbol{\epsilon}^t$ does to $\boldsymbol{0}$. We propose a simple modification of COX-UCB, called ψ-approximated COX-UCB (ψ-COX-UCB), exploiting the faster convergence of the strategy estimation. The only difference between COX-UCB and ψ-COX-UCB is in the strategy selection procedure, as specified in Algorithm 3. This simple variation has a twofold effect on the overall algorithm: (i) in some iterations it avoids solving the bilinear program, by solving a much simpler *linear program* instead, and (ii) it leverages the fast empiric

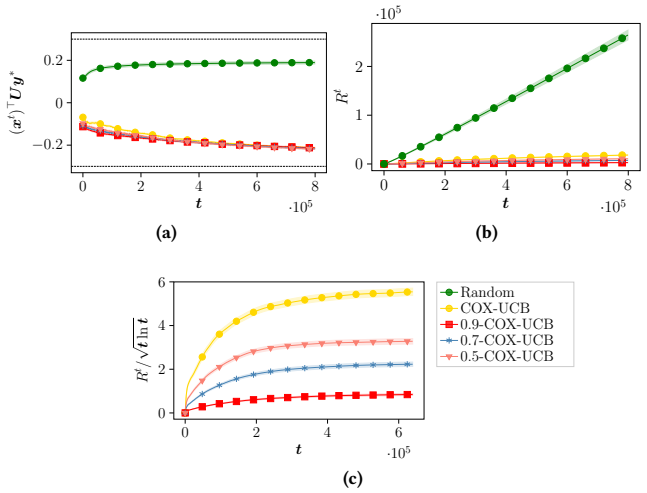

Figure 1: Performances of COX-UCB in Kuhn poker with 7 ranks: player j's utility (top left), cumulative regret (top right), and cumulative regret divided by $\sqrt{t \ln t}$ (bottom).

convergence of $\boldsymbol{y}^t$ to converge faster. In the remaining part of this section, we empirically evaluate COX-UCB and ψ-COX-UCB for different values of the parameter ψ.

6.2 Experimental results

Experimental setting. In order to compare the performances of COX-UCB with those of its approximated version, we consider three versions of ψ-COX-UCB, respectively with $\psi = 0.5$, $\psi = 0.7$ and $\psi = 0.9$. We evaluate the performances of the algorithms against 10 different randomly generated strategies for each game instance considered. For each strategy, we execute 5 different algorithm runs. The values (α, β) needed for the utility constraints are set to $\alpha = -0.3$ and $\beta = 0.3$ in all the experiments. We use Gurobi for solving bilinear optimization problems [19]. As a baseline for comparisons, we use a *random* algorithm that, at each iteration $t \in [T]$, returns a strategy randomly picked from the utility-constrained strategy set $\mathcal{X}^t$.

Convergence results. Figure 1 shows the results of the experiments on *kuhn_7*. We observe that the performances of COX-UCB and those of its approximated versions are comparable, thus showing that the rate of convergence of the opponent's strategy estimation plays a negligible role in this instance. Furthermore, Figure 1a shows how the algorithm manages to maintain the expected utility of the opponent in the utility range $[-0.3, 0.3]$. On the other hand, we can see how the random algorithm, which plays a random strategy at each iteration, does indeed maintain the utility constraints satisfied (as they are below the threshold of $\beta = 0.3$), but it does *not* result in sublinear regret R^T, as one can observe from Figure 1b. Interestingly, in Figure 1c the ratio between the regret and $\sqrt{t \ln t}$, representing its asymptotic dependence on t, converges to a constant as t increases, thus showing that our upper bound on the regret is tight.

7 CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we studied, for the first time, the problem of designing artificial agents that learn how to exploit an unknown human opponent in an online fashion while, at the same time, guaranteeing that the human's expected utility remains bounded within some upper and lower thresholds. This framework finds application in several real-world human-agent interactions, such as repeated human-agent game playing in which the agent's goal is also to keep the human *engaged* in the repeated interaction, and human teaching by means of *serious games*. Our results hold up under the assumption that the human player adopts a fixed (unknown) stochastic strategy. This is a first crucial step towards more complex models of the human behavior. Nevertheless, even this basic case begets considerable technical challenges, which make the theoretical study of the problem interesting in its own, while also constituting a starting point for the analysis of more complex scenarios.

An interesting direction for future research is assuming a more complex human behavior, such as the case in which the human is learning over time. Another line of research is the enhancement of the scalability of the COX-UCB algorithm, so as to enable its application in real-world sequential games. Finally, it could be of general interest framing our algorithm in serious games scenarios, which have been largely ignored by the research community in artificial intelligence so far.

REFERENCES

[1] Amir Zaib Abbasi, Ding Hooi Ting, and Helmut Hlavacs. 2017. Engagement in games: developing an instrument to measure consumer videogame engagement and its validation. *International Journal of Computer Games Technology* 2017 (2017).
[2] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. 2011. Improved Algorithms for Linear Stochastic Bandits.. In *Proceedings of the conference on Neural Information Processing Systems (NeurIPS)*, Vol. 11. 2312–2320.
[3] Stefano V. Albrecht and Subramanian Ramamoorthy. 2014. On Convergence and Optimality of Best-Response Learning with Policy Types in Multiagent Systems. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. 12–21.
[4] Stefano V. Albrecht and Peter Stone. 2017. Reasoning about Hypothetical Agent Behaviours and their Parameters. In *Proceedings of the Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. 547–555.
[5] Peter Auer. 2002. Using Confidence Bounds for Exploitation-Exploration Tradeoffs. *Journal of Machine Learning Research* 3, Nov (2002), 397–422.
[6] Samuel Barrett and Peter Stone. 2015. Cooperating with Unknown Teammates in Complex Domains: A Robot Soccer Case Study of Ad Hoc Teamwork. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*. 2010–2016.
[7] Martino Bernasconi-de-Luca, Federico Cacciamani, Simone Fioravanti, Nicola Gatti, Alberto Marchesi, and Francesco Trovò. 2021. Exploiting Opponents Under Utility Constraints in Sequential Games. *Advances in Neural Information Processing Systems* 34 (2021).
[8] Noam Brown and Tuomas Sandholm. 2018. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science* 359, 6374 (2018), 418–424.
[9] Noam Brown and Tuomas Sandholm. 2019. Superhuman AI for multiplayer poker. *Science* 365, 6456 (2019), 885–890.
[10] Murray Campbell, A.Joseph Hoane, and Feng hsiung Hsu. 2002. Deep Blue. *Artificial Intelligence* 134, 1 (2002), 57–83.
[11] Jeanine A. DeFalco, Jonathan P. Rowe, Luc Paquette, Vasiliki Georgoulas-Sherry, Keith Brawner, Bradford W. Mott, Ryan S. Baker, and James C. Lester. 2018. Detecting and Addressing Frustration in a Serious Game for Military Training. *International Journal of Artificial Intelligence in Education* 28, 2 (2018), 152–193.
[12] Luc Devroye. 1983. The Equivalence of Weak, Strong and Complete Convergence in L1 for Kernel Density Estimates. *The Annals of Statistics* 11, 3 (1983), 896–904.
[13] Ralf Dörner, Stefan Göbel, Wolfgang Effelsberg, and Josef Wiemeyer. 2016. *Serious Games*. Springer.
[14] Attila Egri-Nagy and Antti Törmänen. 2020. The Game Is Not over Yet—Go in the Post-AlphaGo Era. *Philosophies* 5, 4 (2020), 37.
[15] Jakob N. Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. 2018. Learning with Opponent-Learning Awareness. In *Proceedings of Conference on Autonomous Agents and MultiAgent Systems*

(AAMAS). 122–130.
[16] Sam Ganzfried and Tuomas Sandholm. 2011. Game theory-based opponent modeling in large imperfect-information games. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 533–540.
[17] Sam Ganzfried and Tuomas Sandholm. 2015. Safe Opponent Exploitation. *ACM Transaction on Economics and Computations* 3, 2 (2015), 8:1–8:28.
[18] Sam Ganzfried and Qingyun Sun. 2018. Bayesian Opponent Exploitation in Imperfect-Information Games. In *AAAI Spring Symposia*.
[19] LLC Gurobi Optimization. 2021. Gurobi Optimizer Reference Manual. http://www.gurobi.com
[20] He He and Jordan L. Boyd-Graber. 2016. Opponent Modeling in Deep Reinforcement Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, Vol. 48. 1804–1813.
[21] Daphne Koller, Nimrod Megiddo, and Bernhard Von Stengel. 1996. Efficient computation of equilibria for extensive two-person games. *Games and economic behavior* 14, 2 (1996), 247–259.
[22] Harold W Kuhn. 2016. 9. A SIMPLIFIED TWO-PERSON POKER. In *Contributions to the Theory of Games (AM-24), Volume I*. Princeton University Press, 97–104.
[23] Igor Mayer. 2012. Towards a comprehensive methodology for the research and evaluation of serious games. *Procedia Computer Science* 15 (2012), 233–247.
[24] Richard Mealing and Jonathan L. Shapiro. 2017. Opponent Modeling by Expectation-Maximization and Sequence Prediction in Simplified Poker. *IEEE Transaction on Compututational Intelligence AI Games* 9, 1 (2017), 11–24.
[25] Alenka Poplin. 2011. Games and serious games in urban planning: study cases. In *International Conference on Computational Science and Its Applications*. Springer, 1–14.
[26] Rosaldo JF Rossetti, João Emílio Almeida, Zafeiris Kokkinogenis, and Joel Gonçalves. 2013. Playing transportation seriously: Applications of serious games to artificial transportation systems. *IEEE Intelligent Systems* 28, 4 (2013), 107–112.
[27] Yoav Shoham and Kevin Leyton-Brown. 2008. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.
[28] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 7587 (2016), 484–489.
[29] Bernhard Von Stengel. 1996. Efficient computation of behavior strategies. *Games and Economic Behavior* 14, 2 (1996), 220–246.
[30] Ryan Wang, Samuel DeMaria Jr, Andrew Goldberg, and Daniel Katz. 2016. A systematic review of serious games in training health care professionals. *Simulation in Healthcare* 11, 1 (2016), 41–51.
[31] Zhe Wu, Kai Li, Enmin Zhao, Hang Xu, Meng Zhang, Haobo Fu, Bo An, and Junliang Xing. 2021. L2E: Learning to Exploit Your Opponent. *CoRR* abs/2102.09381 (2021).

APPENDIX OF THE PAPER "EXPLOITING OPPONENTS UNDER UTILITY CONSTRAINTS IN SEQUENTIAL GAMES"

The appendix is structured as follows:

- Appendix A provides the proofs omitted from Section 4.1, describing the method adopted for the construction of the confidence region $\mathcal{Y}^{t-1}$ for the human strategy $\boldsymbol{y}^*$.
- Appendix B provides the proofs omitted from Section 4.2, describing the method adopted for the construction of the utility-constrained strategy set $\mathcal{X}^t$ starting from $\mathcal{Y}^{t-1}$.
- Appendix C gives the proof omitted from Section 5 for the regret bound of COX-UCB.
- Appendix D provides some additional experimental results.

A PROOFS OMITTED FROM SECTION 4.1

Lemma 4.2. *Let $\boldsymbol{p} \in \Delta^{|A|}$ and $a^1, \dots, a^t \in A$ be t actions sampled independently according to $\boldsymbol{p}$. Then, for any $0 < \delta \leq 3\exp(-4|A|/5)$, it holds:*

$$\mathbb{P}\left(\bigcap_{a\in A}\left\{\left|\boldsymbol{p}^t[a] - \boldsymbol{p}[a]\right| \leq \frac{5}{2}\sqrt{\frac{\ln(3/\delta)}{t}}\right\}\right) \geq 1-\delta.$$

Proof. Notice that, given the result in Lemma 4.1, it is sufficient to show that, for every $\epsilon > 0$, it holds

$$\sum_{a\in A}\left|\boldsymbol{p}^t[a] - \boldsymbol{p}[a]\right| \leq \epsilon \implies \bigcap_{a\in A}\left\{\left|\boldsymbol{p}^t[a] - \boldsymbol{p}[a]\right| \leq \frac{\epsilon}{2}\right\}.$$

In the following, we prove by contradiction that, if $\left|\boldsymbol{p}^t[a] - \boldsymbol{p}[a]\right| > \frac{\epsilon}{2}$ for some $a \in A$, then $\sum_{a\in A}\left|\boldsymbol{p}^t[a] - \boldsymbol{p}[a]\right| > \epsilon$. Let $\bar{a} \in A$ be such that

$$\epsilon_{\bar{a}} := \boldsymbol{p}^t[\bar{a}] - \boldsymbol{p}[\bar{a}] > \frac{\epsilon}{2}. \tag{4}$$

Then, we have that

$$\sum_{a\in A: a\neq\bar{a}} \boldsymbol{p}^t[a] - \boldsymbol{p}[a] = \sum_{a\in A: a\neq\bar{a}} \boldsymbol{p}^t[a] - \sum_{a\in A: a\neq\bar{a}} \boldsymbol{p}[a] = 1 - \boldsymbol{p}^t[\bar{a}] - 1 + \boldsymbol{p}[\bar{a}] = -\epsilon_{\bar{a}},$$

which, in turn, implies that $\left|\sum_{a\in A: a\neq\bar{a}} \boldsymbol{p}^t[a] - \boldsymbol{p}[a]\right| = \epsilon_{\bar{a}} > \left|\frac{\epsilon}{2}\right|$. Moreover, by the triangular inequality, we have:

$$\sum_{a\in A: a\neq\bar{a}}\left|\boldsymbol{p}^t[a] - \boldsymbol{p}[a]\right| \geq \left|\sum_{a\in A: a\neq\bar{a}} \boldsymbol{p}^t[a] - \boldsymbol{p}[a]\right| > \epsilon/2. \tag{5}$$

By summing Equation (4) and Equation (5), we obtain $\sum_{a\in A}\left|\boldsymbol{p}^t[a] - \boldsymbol{p}[a]\right| > \epsilon$, which is the desired contradiction that proves the result. □

Lemma 4.3. *Let $J \in \mathcal{J}$ be a player j's infoset. Then, for any $0 < \delta \leq 3\exp(-4|A(J)|/5)$ it holds:*

$$\mathbb{P}\Bigg(\bigcap_{a\in A(J)}\Bigg\{\left|\boldsymbol{y}^t[\sigma_j(J)a] - \boldsymbol{y}^*[\sigma_j(J)a]\right| \leq \frac{5}{2\bar{\rho}^t_{-j}(J)}\sqrt{\frac{\ln(3/\delta)}{t}}\Bigg\}\Bigg) \geq 1-\delta.$$

Proof. Since $\boldsymbol{n}^t_J$ follows a multinomial distribution, using Lemma 4.1 provides us with an high-probability confidence region for the components of $\boldsymbol{y}^*$ corresponding to the sequences terminating with an action at infoset J. Formally, since $\boldsymbol{p}^t$ in Lemma 4.1 plays the same role as $\frac{1}{t}\boldsymbol{n}^t_J$, we get:

$$\mathbb{P}\left(\sum_{a\in A(J)}\left|N^t(J,a) - \mathbb{E}[N^t(J,a)]\right| \leq 5\,t\sqrt{\frac{\ln(3/\delta)}{t}}\right) \geq 1-\delta.$$

Let us recall that $\mathbb{E}[N^t(J,a)] = t\,\bar{\rho}^t_{-j}(J)\boldsymbol{y}^*[\sigma_j(J)a]$. Thus, dividing by $t\,\bar{\rho}^t_{-j}(J)$ the argument of the probability in the left hand side of the above equation, we get:

$$\mathbb{P}\left(\sum_{a\in A(J)}\left|\frac{N^t(J,a)}{t\,\bar{\rho}^t_{-j}(J)} - \boldsymbol{y}^*[\sigma_j(J)a]\right| \leq \frac{5}{\bar{\rho}^t_{-j}(J)}\sqrt{\frac{\ln(3/\delta)}{t}}\right) \geq 1-\delta. \tag{6}$$

Following the same line of reasoning of the proof of Lemma 4.2, we conclude that:

$$\mathbb{P}\left(\bigcap_{a\in A(J)}\left\{\left|\frac{N^t(J,a)}{t\bar{\rho}^t_{-j}(J)}-\boldsymbol{y}^*[\sigma_j(J)a]\right|\leq\frac{5}{2\bar{\rho}^t_{-j}(J)}\sqrt{\frac{\ln(3/\delta)}{t}}\right\}\cap E_\diamond\right)\geq 1-\delta, \tag{7}$$

where we define the event $E_\diamond \coloneqq \left\{\left|N^t(J,a_\diamond)-\mathbb{E}\left[N^t(J,a_\diamond)\right]\right|\leq\frac{5t}{2}\sqrt{\frac{\ln(3/\delta)}{t}}\right\}$. The statement follows from the fact that, for two generic events E and E', it holds $\mathbb{P}(E\cap E')\leq\mathbb{P}(E)$. □

Theorem 4.4. *For every player j's infoset $J\in\mathcal{J}$, let $\delta_J\in(0,1)$ be such that the condition in Lemma 4.3 is satisfied and $\sum_{J\in\mathcal{J}}\delta_J<1$. Then, $\mathbb{P}\left(\boldsymbol{y}^*\in\mathcal{Y}^t\right)\geq 1-\delta$, where $\delta\coloneqq\sum_{J\in\mathcal{J}}\delta_J$ and*

$$\mathcal{Y}^t\coloneqq\Bigg\{\boldsymbol{y}\in\mathcal{Y}:\left|\boldsymbol{y}^t[\sigma_j(J)a]-\boldsymbol{y}^*[\sigma_j(J)a]\right|\leq \frac{5}{2\bar{\rho}^t_{-j}(J)}\sqrt{\frac{\ln(3/\delta_J)}{t}}\quad\forall J\in\mathcal{J},\forall a\in A(J)\Bigg\}.$$

Proof. For each infoset $J\in\mathcal{J}$, let us apply Lemma 4.3 with $\delta=\delta_J\leq 3\exp(-4|A(J)|/5)$. The lemma states that for each $J\in\mathcal{J}$, the event

$$E_J\coloneqq\bigcap_{a\in A(J)}\left\{\left|\boldsymbol{y}^t[\sigma_j(J)a]-\boldsymbol{y}^*[\sigma_j(J)a]\right|\leq\frac{5}{2\bar{\rho}^t_{-j}(J)}\sqrt{\frac{\ln(3/\delta_J)}{t}}\right\} \tag{8}$$

holds with probability at least $1-\delta_J$. By applying a union bound, we have that:

$$\begin{aligned}\mathbb{P}\left(\bigcap_{J\in\mathcal{J}}E_J\right)&=1-\mathbb{P}\left(\bigcup_{J\in\mathcal{J}}\neg E_J\right)\\&\geq 1-\sum_{J\in\mathcal{J}}1-\mathbb{P}\left(E_J\right)\\&\geq 1-\sum_{J\in\mathcal{J}}\delta_J\\&=1-\delta.\end{aligned}$$

Finally, choosing the errors δ_J such that $\sum_{J\in\mathcal{J}}\delta_J<1$ proves the result. □

B PROOFS OMITTED FROM SECTION 4.2

Theorem 4.5. *Let $t\in[T]$ and $\delta\in(0,1)$. Given $\alpha\in\mathbb{R}$ and $\beta\in\mathbb{R}$, it holds:*

$$\begin{aligned}\mathcal{X}^t=\mathcal{X}\cap\big\{(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\omega}):\boldsymbol{u},\boldsymbol{\omega}\geq\boldsymbol{0},\ \boldsymbol{b}^\top\boldsymbol{u}\leq\beta,\ \boldsymbol{A}^\top\boldsymbol{u}\geq\boldsymbol{U}_j^\top\boldsymbol{x},\\ -\boldsymbol{b}^\top\boldsymbol{\omega}\geq\alpha,\ -\boldsymbol{A}^\top\boldsymbol{\omega}\leq\boldsymbol{U}_j^\top\boldsymbol{x}\big\}.\end{aligned}$$

Proof. The proof follows the reasoning outlined in Section 4.2. First, we notice that $\boldsymbol{x}\in\mathcal{X}$ belongs to the utility-constrained strategy set $\mathcal{X}^t$ at iteration $t\in[T]$ if and only if

$$\max_{\boldsymbol{y}\in\mathcal{Y}^{t-1}}\boldsymbol{x}^\top\boldsymbol{U}_j\boldsymbol{y}\leq\beta\quad\wedge\quad\min_{\boldsymbol{y}\in\mathcal{Y}^{t-1}}\boldsymbol{x}^\top\boldsymbol{U}_j\boldsymbol{y}\geq\alpha.$$

By first considering the max problem, we can write it as the linear program in Problem (3) in the main paper. Then, its dual problem reads as follows:

$$\min_{\boldsymbol{u}\geq\boldsymbol{0}}\boldsymbol{b}^\top\boldsymbol{u}\quad\text{s.t.} \tag{9a}$$

$$\boldsymbol{A}^\top\boldsymbol{u}\geq\boldsymbol{U}_j^\top\boldsymbol{x}, \tag{9b}$$

where $\boldsymbol{u}\in\mathbb{R}^{(|\mathcal{J}|+1)\times 2|\Sigma_j|}$ is a vector of dual variables, while:

$$\boldsymbol{A}\coloneqq\begin{bmatrix}\boldsymbol{I}_{|\Sigma_j|}\\-\boldsymbol{I}_{|\Sigma_j|}\\\boldsymbol{F}_j\\-\boldsymbol{F}_j\end{bmatrix}\quad\text{and}\quad\boldsymbol{b}\coloneqq\begin{bmatrix}\boldsymbol{y}^{t-1}+\boldsymbol{\epsilon}^{t-1}\\-\boldsymbol{y}^{t-1}+\boldsymbol{\epsilon}^{t-1}\\\boldsymbol{f}_j\\-\boldsymbol{f}_j\end{bmatrix},$$

with $\boldsymbol{I}_n$ being the $n\times n$ identity matrix. By strong duality, the optimal dual objective equates the optimal primal objective, and, thus, given any player i's strategy $\boldsymbol{x}\in\mathcal{X}$, the condition $\max_{\boldsymbol{y}\in\mathcal{Y}^{t-1}}\boldsymbol{x}^\top\boldsymbol{U}_j\boldsymbol{y}\leq\beta$ holds if there exists a dual feasible solution $\boldsymbol{u}$ that satisfies the additional

constraint that $\boldsymbol{b}^\top \boldsymbol{u} \leq \beta$. Following an analogous reasoning for $\min_{\boldsymbol{y} \in \mathcal{Y}^{t-1}} \boldsymbol{x}^\top \boldsymbol{U}_j \boldsymbol{y}$, and letting $\boldsymbol{\omega} \in \mathbb{R}^{(|\mathcal{J}|+1)\times 2|\Sigma_j|}$ be a vector of variables of the dual problem corresponding to its linear programming formulation (similar to Problem (3)), the result follows. □

Discussion on the emptiness of $\mathcal{X}^t$ in Theorem 4.5. In some cases, the set $\mathcal{X}^t$ defined in Theorem 4.5 could be empty. This happens when the components of the vector $\boldsymbol{\epsilon}^t$ are large, as it is usually the case after the first game repetitions. However, since by assumption the problem is feasible for the true sequence-form strategy $\boldsymbol{y}^*$, then the set $\mathcal{X}^t$ will be non-empty after a finite number of iterations. Hence, as customary in safe exploration problems, we can assume that we have at our disposal an initial number of plays that allow to have an estimate of $\boldsymbol{y}^*$ that is good enough (*i.e.*, with a small norm of $\boldsymbol{\epsilon}^t$) so that $\mathcal{X}^t$ is non-empty. In practice, one does *not* need to wait that $\mathcal{X}^t$ is always non-empty, and can mix the initial pure-exploration phase with the selection strategy implemented by the algorithm. For instance, this can be achieved by playing a random strategy when $\mathcal{X}^t$ is empty, while following the algorithm recommendation when $\mathcal{X}^t$ is non-empty. In the experimental evaluation, which is discussed in details in Appendix D, we use the variable `N_BLANK_GAMES` to tune this aspect of the algorithm implementation.

C PROOFS OMITTED FROM SECTION 5

Before proving Theorem 5.1, we need to show the following technical lemma.

Lemma C.1. *Let $f(\tau) := \frac{2\ln^2 \tau + \ln \tau + 1}{\sqrt{\ln \tau}(\ln \tau + 1)^2}$. Then, it holds that:*

$$\sum_{\tau=1}^{t} f(\tau) \geq \frac{2t\sqrt{\ln t}}{\ln t + 1}. \tag{10}$$

Proof. By noticing that $f(\tau)$ is decreasing in τ, we can use the following integral inequality:

$$\sum_{\tau=1}^{t} f(\tau) \geq \sum_{\tau=1}^{t} \int_{\tau}^{\tau+1} f(x)dx = \int_{1}^{t+1} f(x)dx \geq \int_{1}^{t} f(x)dx = \frac{2t\sqrt{\ln t}}{\ln t + 1},$$

which shows the result. □

Theorem 5.1. *Let $\alpha^t := \eta \frac{2\ln^2 t + \ln t + 1}{\sqrt{\ln t}\,(\ln t + 1)^2}$ for every $t \in [T]$, where $\eta \in (0,1)$, and let $\delta \in (0,1)$. The COX-UCB algorithm attains the following regret bound with probability at least $1-\delta$:*

$$R^T \leq \frac{5}{2\eta} K_{\boldsymbol{U}_i} C \left(1 + 2\sqrt{T \ln T}\right),$$

where $K_{\boldsymbol{U}_i} := ||\boldsymbol{U}_i||_\infty$ and C is a suitably-defined constant.

Proof. First, let us recall that the confidence regions $\mathcal{Y}^{t-1}$ used by the COX-UCB algorithm are built by applying Theorem 4.4 with error tolerances $\delta_J \in (0,1)$, for $J \in \mathcal{J}$, such that the conditions in Theorem 4.4 are satisfied and $\delta = \sum_{J \in \mathcal{J}} \delta_J$. In the following, we prove the desired regret bound by bounding the regret that player i suffers ar each iteration t.

For every $t \in [T]$, we let $\boldsymbol{x}^{t,*} \in \operatorname{argmax}_{\boldsymbol{x}^* \in \mathcal{X}^t} (\boldsymbol{x}^*)^\top \boldsymbol{U}_i \boldsymbol{y}^*$. Then, at each iteration t, player i incurs in an instantaneous regret r^t, which is formally defined as follows:

$$r^t := (\boldsymbol{x}^{t,*})^\top \boldsymbol{U}_i \boldsymbol{y}^* - (\boldsymbol{x}^t)^\top \boldsymbol{U}_i \boldsymbol{y}^*.$$

Since the COX-UCB algorithm selects strategies $\boldsymbol{x}^t$ so that $\boldsymbol{x}^t \in \operatorname{argmax}_{\boldsymbol{x} \in \tilde{\mathcal{X}}^t} \max_{\boldsymbol{y} \in \mathcal{Y}^{t-1}} \boldsymbol{x}^\top \boldsymbol{U}_i \boldsymbol{y}$ (see Algorithm 2), we have that, with probability at least $1-\delta$, it holds

$$r_t \leq (\boldsymbol{x}^t)^\top \boldsymbol{U}_i \tilde{\boldsymbol{y}}^t - (\boldsymbol{x}^t)^\top \boldsymbol{U}_i \boldsymbol{y}^*, \tag{11}$$

where we let $\tilde{\boldsymbol{y}}^t \in \operatorname{argmax}_{\boldsymbol{y} \in \mathcal{Y}^{t-1}} (\boldsymbol{x}^t)^\top \boldsymbol{U}_i \boldsymbol{y}$. By using the definition of the sequence-form utility matrix $\boldsymbol{U}_i$, we can re-write Equation (11) as follows:

$$r_t \leq (\boldsymbol{x}^t)^\top \boldsymbol{U}_i (\tilde{\boldsymbol{y}}^t - \boldsymbol{y}^*) = \sum_{z \in Z} \boldsymbol{x}^t[\sigma_i(z)]\, \boldsymbol{U}_i[\sigma_i(z), \sigma_j(z)] \left(\tilde{\boldsymbol{y}}^t[\sigma_j(z)] - \boldsymbol{y}^*[\sigma_j(z)]\right).$$

For every terminal node $z \in Z$, by letting $J(z) \in \mathcal{J}$ be the (unique w.l.o.g.) infoset such that the last action of the sequence $\sigma_j(z)$ is played at $J(z)$, we can invoke Theorem 4.4 together with the Cauchy-Swartz inequality to obtain that, with probability at least $1 - \delta_{J(z)}$, the following holds

$$\tilde{\boldsymbol{y}}^t[\sigma_j(z)] - \boldsymbol{y}^*[\sigma_j(z)] \leq \frac{5}{\bar{\rho}^t_{-j}(J(z))} \sqrt{\frac{\ln(3/\delta_{J(z)})}{t}}. \tag{12}$$

Moreover, since by definition of $\tilde{\mathcal{X}}^t$, we have that $\boldsymbol{x}^t[\sigma_i] \geq \alpha^t$ for all $\sigma_i \in \Sigma_i$. This gives us the following lower bound on the probability $\rho^t_{-j}(J(z))$:

$$\rho^t_{-j}(J(z)) = \sum_{h \in J(z)} \boldsymbol{x}^t[\sigma_i(h)] p_c(h) \geq \alpha^t \sum_{h \in J(z)} p_c(h) = p_c(J(z))\alpha^t, \tag{13}$$

where we let $p_c(J(z)) \coloneqq \sum_{h \in J(z)} p_c(h)$.

Then, the term $\frac{1}{\bar{\rho}^t_{-j}(J_z)\sqrt{t}}$ in Equation (12) can be bounded as follows.

$$\frac{1}{\bar{\rho}^t_{-j}(J(z))\sqrt{t}} = \frac{\sqrt{t}}{\sum_{\tau=1}^{t} \rho^t_{-j}(J(z))} \leq \frac{\sqrt{t}}{p_c(J(z)) \sum_{\tau=1}^{t} \alpha^\tau} \leq \frac{1}{2\eta p_c(J(z))} \frac{\ln t + 1}{\sqrt{t \ln t}}, \tag{14}$$

where the first inequality follows from Equation (13) and the second one comes from Lemma C.1.

Therefore, by combining Equation 12 and Equation (14), we obtain:

$$\tilde{\boldsymbol{y}}^t[\sigma_j(z)] - \boldsymbol{y}^*[\sigma_j(z)] \leq \frac{5\sqrt{\ln(3/\delta_{J(z)})}}{2\eta p_c(J(z))} \frac{\ln t + 1}{\sqrt{t \ln t}}, \tag{15}$$

which holds with probability at least $1 - \delta_{J(z)}$.

Using Equation (15) and observing that $\boldsymbol{x}^t[\sigma_i(z)] \leq 1$ for all $z \in Z$, we can conclude that, with probability at least $1 - \delta$, it holds:

$$R_T \coloneqq \sum_{t=1}^{T} r_t \leq \sum_{t=1}^{T} (\boldsymbol{x}^t)^\top \boldsymbol{U}_i (\tilde{\boldsymbol{y}}^t - \boldsymbol{y}^*) \tag{16}$$

$$\leq K_{\boldsymbol{U}_i} \sum_{t=1}^{T} \sum_{z \in Z} \frac{5\sqrt{\ln(3/\delta_{J(z)})}}{2\eta p_c(J(z))} \frac{\ln t + 1}{\sqrt{t \ln t}} \tag{17}$$

$$= K_{\boldsymbol{U}_i} \frac{5}{2\eta} \left(\sum_{z \in Z} \frac{\sqrt{\ln(3/\delta_{J(z)})}}{p_c(J(z))} \right) \sum_{t=1}^{T} \frac{\ln t + 1}{\sqrt{t \ln t}} \tag{18}$$

$$\leq K_{\boldsymbol{U}_i} \frac{5}{2\eta} \left(\sum_{z \in Z} \frac{\sqrt{\ln(3/\delta_{J(z)})}}{p_c(J(z))} \right) \left(1 + \sum_{t=1}^{T-1} \int_{\tau=t}^{t+1} \frac{\ln \tau + 1}{\sqrt{\tau \ln \tau}} d\tau \right) \tag{19}$$

$$= K_{\boldsymbol{U}_i} \frac{5}{2\eta} \left(\sum_{z \in Z} \frac{\sqrt{\ln(3/\delta_{J(z)})}}{p_c(J(z))} \right) \left(1 + \int_{2}^{T} \frac{\ln \tau + 1}{\sqrt{\tau \ln \tau}} d\tau \right) \tag{20}$$

$$\leq K_{\boldsymbol{U}_i} \frac{5}{2\eta} \left(\sum_{z \in Z} \frac{\sqrt{\ln(3/\delta_{J(z)})}}{p_c(J(z))} \right) \left(1 + 2\sqrt{T \ln T} \right), \tag{21}$$

where we let $K_{\boldsymbol{U}_i} \coloneqq ||\boldsymbol{U}_i||_\infty$ and $C \coloneqq \sum_{z \in Z} \frac{\sqrt{\ln(3/\delta_{J(z)})}}{p_c(J(z))}$. This concludes the proof. □

Theorem 5.1 gives a sublinear upper bound on the regret of the COX-UCB algorithm. Notice that the order of the regret is $\sqrt{T \ln T}$ and that the constant C is linear in the number of terminal nodes Z.

D ADDITIONAL DETAILS ON THE EXPERIMENTAL EVALUATION

In this section, we provide additional details and results on the experimental evaluation of our COX-UCB and ψ-COX-UCB algorithms. We test the two algorithms in three different instances of Kuhn poker with ranks 3, 5 and 7 (denoted respectively as *kuhn_3*, *kuhn_5* and *kuhn_7*) and in one instance of Leduc poker with ranks 2 (denoted as *leduc_2*).

D.1 Experimental setting and hyperparameters

In order to guarantee that the utility-constrained strategy set $\mathcal{X}^t$ is non-empty at the beginning of the repeated interaction, we assume to have access to some prior information on the strategy employed by the human player, so as to reduce the initial uncertainty encoded by the confidence region $\mathcal{Y}^{t-1}$. This is reasonable in practice, since a new player can always be profiled according to a number of user classes. In particular, we encode this information as observations collected during a number `N_BLANK_GAMES` of games played by the players at the beginning of the repeated interaction, in which the agent player adopts a purely-explorative strategy.

Furthermore, since the estimation and bounds do not change significantly after a single game, we compute the solution to the optimization problem required by COX-UCB every `UPDATE_EVERY` iterations. Moreover, we set a time limit (`TIME_LIMIT`) to the Gurobi solver to solve the bilinear program. This allows us to reduce significantly the time spent to solve bilinear optimization problems.

In all our experiments, the values of the hyperparameters are set to:

- $\delta = 0.05$ and $\delta_J = \delta/|\mathcal{J}|$ for all $J \in \mathcal{J}$;
- N_BLANK_GAMES = 1000;
- UPDATE_EVERY = 20;
- TIME_LIMIT = $1s$;
- $\eta = 0.05/|\Sigma_i|$;
- utility constraints lower bound $\alpha = -0.3$;
- utility constraints upper bound $\beta = 0.3$.

We fix the number of iterations after which we stop the execution of our algorithms to $2e5$, $4e5$, $8e5$ and $2e6$ for *kuhn_3*, *kuhn_5*, *kuhn_7* and *leduc_2*, respectively.

Finally, the infrastructure used to run the experiments is a 32-core UNIX system with 128 GB RAM.

D.2 Detailed experimental results

Figure 2 shows the performances of COX-UCB and ψ-COX-UCB. The values tested for the hyperparameter ψ are $\psi = 0.5$, $\psi = 0.7$ and $\psi = 0.9$. As a baseline we use a random policy that consists in randomly selecting a sequence-form strategy from the set $\mathcal{X}^t$ at every time step t. The first column of Figure 2 shows the expected utility of the opponent over the iterations of the algorithm. As we can observe, empirically, the random policy satisfies the utility constraints. This is reasonable, since the strategies are selected from the interior of the utility-constrained strategy set $\mathcal{X}^t$. However, in all the game instances considered, only COX-UCB and ψ-COX-UCB approach the optimal utility values, which are shown in Table 1. Looking at the plots of the cumulative regret (second column of Figure 2), we can observe that COX-UCB and ψ-COX-UCB achieve a significantly lower regret than the baseline. The experiments on *leduc_2* remark the relevance of the convergence rate of the confidence bound on the opponent's strategy. In particular, when the strategy space is larger—as it is the case of *leduc_2*—, the fact that the confidence bound reduces slowly causes a decrease in the performances, slowing down the convergence to an optimal strategy. In this scenario, the approximation yielded by ψ-COX-UCB allows the algorithm to exploit the faster empirical convergence rate of the average strategy, thus resulting in a lower cumulative regret. Finally, the plots on the third column of Figure 2 allow us to evaluate the upper bound derived for the cumulative regret. In particular, we point out that the ratio between the cumulative regret and $\sqrt{t \ln(t)}$ converges to an horizontal line, meaning that the bounds that we derived are tight.

Table 1: Optimal expected utility for the opponent in *kuhn_3*, *kuhn_5*, *kuhn_7*, and *leduc_2*.

-	$-\max_{\boldsymbol{x} \in \mathcal{X}^\star} \boldsymbol{x}^\top \boldsymbol{U}_i \boldsymbol{y}^\star$
kuhn_3	-0.28
kuhn_5	-0.3
kuhn_7	-0.29
leduc_2	-0.3

Table 2: Average time per iteration for the algorithms COX-UCB, 0.5-COX-UCB, 0.7-COX-UCB and 0.9-COX-UCB in *kuhn_3*, *kuhn_5*, *kuhn_7*, and *leduc_2*.

-	COX-UCB	0.5-COX-UCB	0.7-COX-UCB	0.9-COX-UCB
kuhn_3	$0.011s$	$0.006s$	$0.004s$	$0.003s$
kuhn_5	$0.011s$	$0.006s$	$0.004s$	$0.004s$
kuhn_7	$0.012s$	$0.007s$	$0.005s$	$0.005s$
leduc_2	$0.016s$	$0.009s$	$0.007s$	$0.006s$

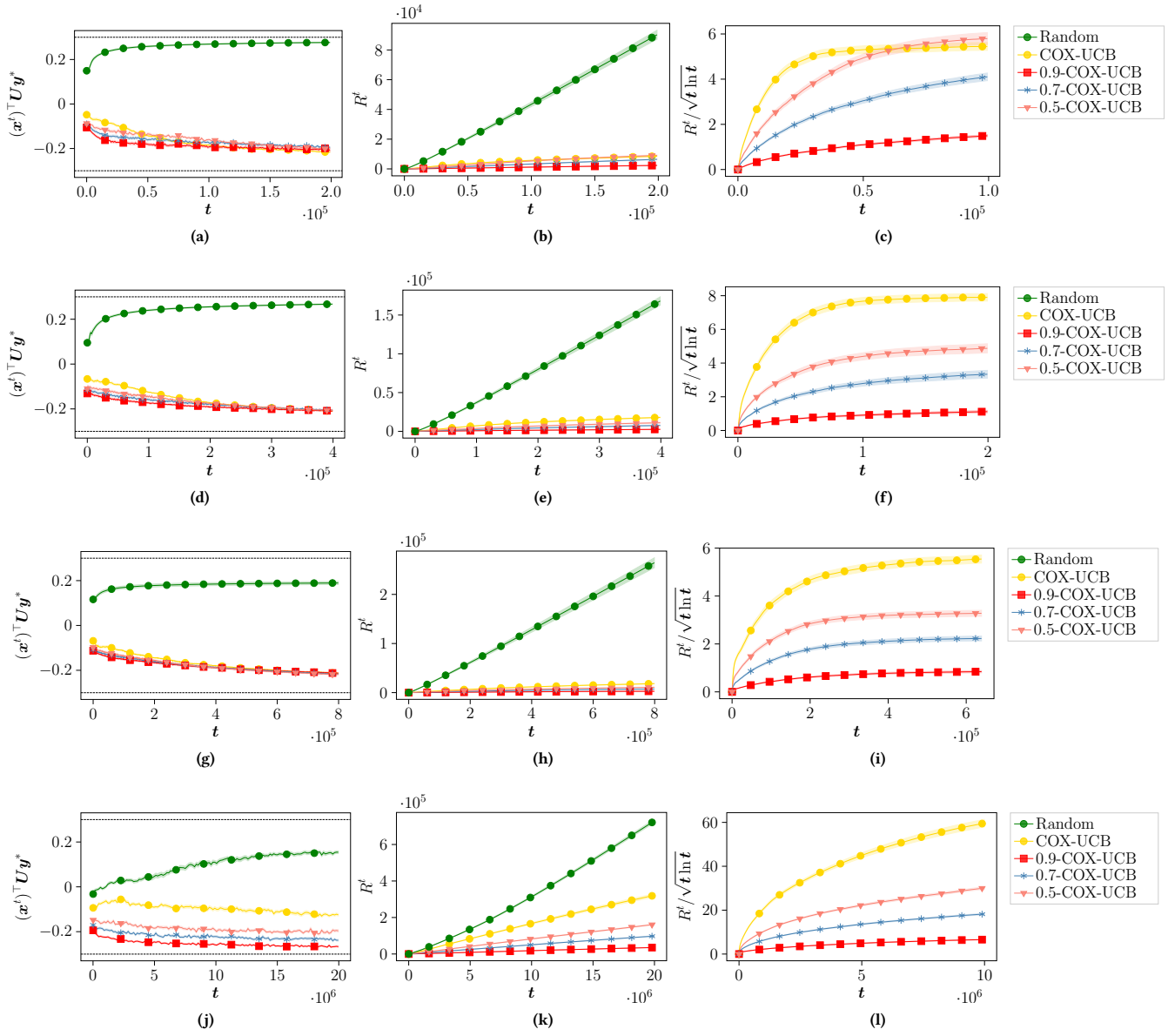

Figure 2: Performances of COX-UCB in Kuhn poker with 3 (top row), 5 (second row) and 7 (third row) ranks and in Leduc poker with 2 ranks (bottom row). From left to right: player j's utility, cumulative regret, and cumulative regret divided by $\sqrt{t \ln t}$.