Simple Uncoupled No-regret Learning Dynamics for Extensive-form Correlated Equilibrium

GABRIELE FARINA, Carnegie Mellon University, USA
ANDREA CELLI, Bocconi University, Italy
ALBERTO MARCHESI and NICOLA GATTI, Politecnico di Milano, Italy

The existence of simple uncoupled no-regret learning dynamics that converge to correlated equilibria in normal-form games is a celebrated result in the theory of multi-agent systems. Specifically, it has been known for more than 20 years that when all players seek to minimize their *internal* regret in a repeated normal-form game, the empirical frequency of play converges to a normal-form correlated equilibrium. Extensive-form (that is, tree-form) games generalize normal-form games by modeling both sequential and simultaneous moves, as well as imperfect information. Because of the sequential nature and presence of private information in the game, correlation in extensive-form games possesses significantly different properties than in normal-form games, many of which are still open research directions. Extensive-form correlated equilibrium (EFCE) has been proposed as the natural extensive-form counterpart to the classical notion of correlated equilibrium in normal-form games. Compared to the latter, the constraints that define the set of EFCEs are significantly more complex, as the correlation device (a.k.a. mediator) must take into account the evolution of beliefs of each player as they make observations throughout the game. Due to that significant added complexity, the existence of uncoupled learning dynamics leading to an EFCE has remained a challenging open research question for a long time. In this article, we settle that question by giving the first uncoupled no-regret dynamics that converge to the set of EFCEs in n-player general-sum extensive-form games with perfect recall. We show that each iterate can be computed in time polynomial in the size of the game tree, and that, when all players play repeatedly according to our learning dynamics, the empirical frequency of play after T game repetitions is proven to be a $O(1/\sqrt{T})$-approximate EFCE with high probability, and an EFCE almost surely in the limit.

CCS Concepts: • **Theory of computation** → **Convergence and learning in games**; *Exact and approximate computation of equilibria;*

Additional Key Words and Phrases: Extensive-form games, correlated equilibrium, regret minimization, multi-agent learning

1 INTRODUCTION

The **Nash equilibrium (NE)** [38] is the most common notion of rationality in game theory, and its
computation in two-player zero-sum games has been the flagship computational challenge at the
interface between computer science and game theory (see, e.g., the landmark results in heads-up
no-limit poker by Brown and Sandholm [3] and Moravčík et al. [35]). The assumption underpin-
ning NE is that the interaction among players is fully *decentralized*. Therefore, an NE is an element
of the *uncorrelated* strategy space of the game, that is, a product of independent probability dis-
tributions over actions, one per player. A competing notion of rationality is the **correlated equi-
librium (CE)** proposed by Aumann [1]. A CE is defined as a probability distribution over joint
action profiles—specifying an action for each player—and it is customarily modeled via a trusted
external *mediator* that draws an action profile from this distribution and privately recommends to
each player their component. The probability distribution is a CE if no player has an incentive to
choose an action different from the mediator's recommendation, because, assuming that all other
players follow their recommended action, the suggested action is the best in expectation.

Many real-world strategic interactions involve more than two players with arbitrary (i.e.,
general-sum) utilities. In those settings, the CE is an appealing solution concept, as it overcomes
several weaknesses of the NE. First, the NE is prone to equilibrium selection issues, raising the
question as to how players can select an equilibrium while they are assumed not to be able to
communicate with each other. Second, computing an NE is computationally intractable, being
PPAD-complete even in two-player games [10, 11], whereas a CE can be computed in polynomial
time.[1] Third, the social welfare that can be attained by an NE may be arbitrarily lower than what
can be achieved via a CE [6, 31, 42]. Last, in normal-form (that is, simultaneous-move) games, the
notion of CE arises from simple *uncoupled* learning dynamics even in general-sum settings with an
arbitrary number of players. In words, these learning dynamics are such that each player adjusts
their strategy on the basis of their own payoff function, and on other players' strategies, but not
on the payoff functions of other players. The existence of uncoupled dynamics allows to overcome
the—often unreasonable—assumption that players have perfect knowledge of other players' pay-
off functions, while at the same time offering a parallel, scalable avenue for finding equilibria. In
contrast, in the case of the NE, uncoupled learning dynamics are only known in the two-player
zero-sum setting [9, 22, 24] or in multi-player games with particular structures (see, e.g., the case
of *polymatrix* games [12]). All of the above considerations contribute to the idea that CE is often
a better prescriptive solution concept than NE in general-sum and multi-player settings.

Extensive-form correlated equilibrium (EFCE), introduced by von Stengel and Forges [48],
is a natural extension of the correlated equilibrium to the case of extensive-form (that is, tree-
form, sequential) games. Extensive-form games generalize normal-form games by modeling both
sequential and simultaneous moves, as well as imperfect information. In an EFCE, the mediator
draws, before the beginning of the sequential interaction, a recommended action for each of the
possible decision points (also known as *information sets*) that players may encounter in the game,

---

[1]In normal-form games, a CE can be computed in polynomial time via linear programming. In extensive-form games, the
computational complexity of computing a CE depends on the specific notion of correlation that is adopted. As discussed
in more detail in the following, the problem can be solved in polynomial time for the notion studied in this article.

but these recommendations are not immediately revealed to each player. Instead, the mediator incrementally reveals relevant individual moves as players reach new information sets. At any decision point, the acting player is free to deviate from the recommended action, but doing so comes at the cost of future recommendations, which are no longer issued to that player if they deviate. It is up to the mediator to make sure that the recommended behavior is indeed an equilibrium—that is, that no player would be better off ever deviating from following the mediator's recommendations at each information set. Compared to the constraints that characterize the set of CEs in normal-form games, those that define the set of EFCEs in extensive-form games are significantly more complex. Indeed, the main challenge of the EFCE case is that the mediator must take into account the evolution of beliefs of each player as they make observations throughout the game tree.

One could define a CE for an extensive-form game by allowing the mediator to draw and recommend an action for each information set to each player before the game starts. Then, each player could decide whether to follow the recommendation or deviate to an arbitrary strategy they desire. In an EFCE, players know less about the action recommendations that were sampled by the mediator than in a CE for extensive-form games, where the whole set of recommendations is immediately revealed. Therefore, by exploiting an EFCE, the mediator can more easily incentivize players to follow strategies that may *hurt* them, as long as players are indifferent as to whether or not to follow the recommendations. This is beneficial when the mediator wants to maximize, e.g., the social-welfare of the game.

In general-sum extensive-form games with an arbitrary number of players (including potentially the *chance player* modeling exogenous stochastic events), the problem of computing a feasible EFCE can be solved in polynomial time in the size of the game tree [27] via a variation of the *Ellipsoid Against Hope* algorithm [28, 40]. Dudík and Gordon [13] provide an alternative sampling-based algorithm to compute EFCEs. However, their algorithm is centralized and based on MCMC sampling, which limits its applicability on large-scale problems. In practice, these approaches cannot scale beyond toy problems. However, methods based on uncoupled learning dynamics usually work quite well in large real-world problems, while retaining the appealing properties of uncoupled dynamics that we discussed above.

The following fundamental research question remains open:

> *Is it possible to devise uncoupled learning dynamics that converge to an EFCE?*

We show that the answer is positive, at least in the full-information feedback model.

In the first part of the article, we formalize the notion of *trigger regret*, simplifying and extending an idea by Gordon et al. [20]. Trigger regret is a notion of regret suitable for extensive-form games that naturally expresses the regret incurred by each player for following the recommendations issued by the EFCE mediator, instead of deviating according to some optimal-in-hindsight strategy. Specifically, trigger regret is a particular instantiation of the framework known as *phi-regret minimization* introduced by Stoltz and Lugosi [44] building on previous work by Greenwald and Jafari [21]. In general, phi-regret minimization operates with a notion of regret defined with respect to a given set of linear transformations on the decision set. To define trigger regret, we identify suitable linear transformations that allow us to encode the behavior of trigger agents in the definition of EFCE, which we coin *canonical trigger deviation functions*. Intuitively, canonical trigger deviation functions encode all the possible ways in which a trigger agent may deviate from the recommendations issued by the EFCE mediator, and instead start playing from that point on according to a different strategy than the recommended one. Our core result on trigger regret is the following: If each player plays according to a no-trigger-regret learning algorithm, then the empirical frequency of play approaches the set of EFCEs.

In the rest of the article, we provide an efficient (that is, requiring time polynomial in the size of the game tree at each iteration) algorithm that minimizes trigger regret. The algorithm is based on the general template for constructing phi-regret minimization algorithms given by Gordon et al. [20], extending prior work by Hazan and Kale [25]. Before one can use that template, two missing pieces need to be solved: (1) constructing an efficient regret minimizer for the set of all valid canonical trigger deviation functions, and (2) showing that any convex combination of canonical trigger deviation functions admits a fixed point strategy, and that such fixed point can be computed efficiently. We solve (1) by exploiting the non-trivial combinatorial structure of the set of canonical trigger deviation functions, and (2) by giving an efficient incremental procedure to compute the fixed point strategy in a top-down traversal of the game tree. Our resulting algorithm minimizes trigger regret, guaranteeing $O(\sqrt{T})$ trigger regret with high probability after T iterations and requiring time polynomial in the size of the game tree at each iteration. Thus, when all players play according to the uncoupled learning dynamics defined by our algorithm, the empirical frequency of play after T game repetitions is proven to be a $O(1/\sqrt{T})$-approximate EFCE with high probability, and an EFCE almost surely in the limit. These results generalize the seminal work by Hart and Mas-Colell [22] to the extensive-form game case via a simple and natural framework.

1.1 Related Work

The study of adaptive procedures leading to a CE dates back to at least the seminal works by Foster and Vohra [16], Fudenberg and Levine [17, 19], and Hart and Mas-Colell [22, 23]; see also the monograph by Fudenberg and Levine [18]. In particular, the work by Hart and Mas-Colell [22] proves that simple dynamics based on the notion of *internal regret* result in empirical frequencies of play that converge to the set of CEs in normal-form games. The strategy that the authors introduce—the so-called *regret matching*—is conceptually simple and guarantees that if all players follow this strategy, then the empirical frequency of play converges to the set of CEs (see also Cahn [5]). Other works describe extensions to the models studied in the aforementioned papers. For example, Stoltz and Lugosi [44] describe an adaptive procedure such that the resulting empirical frequency of play converges to the set of CEs in games with an infinite, but compact, set of actions, while Kakade et al. [29] consider efficient algorithms for computing correlated equilibria in graphical games.

In more recent years, a growing effort has been devoted to understanding the relationships between no-regret learning dynamics and equilibria in extensive-form games. These games pose additional challenges when compared to normal-form games, due to their sequential nature and the presence of imperfect information. While in two-player zero-sum extensive-form games it is widely known that no-regret learning dynamics converge to an NE—with the **counterfactual regret minimization (CFR)** algorithm and its variations being the state-of-the-art for equilibrium finding in such games [4, 33, 45, 46, 49]—the general case is less understood. Celli et al. [7] provide some variations of the classical CFR algorithm for n-player general-sum extensive-form games, showing that they provably converge to a *normal-form coarse correlated equilibrium*, which is based on a form of correlation that is less appealing than that of EFCE in sequential games. Indeed, normal-form coarse correlated equilibria require that the players commit to following all the recommendations issued by the mediator upfront before the beginning of the game, which is not realistic in practice.

Finally, we mention relevant literature subsequent to the conference version of this article. In a recent paper, Morrill et al. [37] conduct a study of different forms of correlation in extensive-form games, defining a taxonomy of solution concepts. Each of their solution concepts is attained by a particular set of no-regret learning dynamics, which is obtained by instantiating the phi-regret minimization framework [20, 21, 44] with a suitably defined deviation function. As part of their

analysis, Morrill et al. [37] investigate some properties of the well-established CFR regret mini-mization algorithm [49] applied to n-player general-sum extensive-form games, establishing that it is hindsight-rational with respect to a specific set of deviation functions, which the authors coin *blind counterfactual deviations*. In subsequent recent work, Morrill et al. [36] extend their prior work [37] by identifying a general class of deviations—called *behavioral deviations*—that induce equilibria that can be found through uncoupled no-regret learning dynamics. Behavioral devia-tions are defined as those specifying an action transformation independently at each information set of the game. As the authors note, the deviation functions involved in the definition of EFCE do not fall under that category. A particular class of behavioral deviation functions—called *causal partial sequence deviations*—induces solution concepts that are (subsets of) EFCEs. Thus, their re-sult begets an alternative set of no-regret learning dynamics that converge to EFCE, based on a different set of deviation functions than those we use in this article.

2 PRELIMINARIES

In this section, we provide some standard definitions related to extensive-form games and regret minimization that will be employed in the remainder of the article. A more comprehensive treat-ment of basic concepts in the theory of extensive-form games can be found in the book by Shoham and Leyton-Brown [43], and an introduction to the theory of learning in games can be found in the book by Cesa-Bianchi and Lugosi [9].

2.1 Mathematical Notation and Algorithmic Conventions

In this article, we adopt the following notational and algorithmic conventions:

- We denote the set of real numbers as $\mathbb{R}$, the set of nonnegative real numbers as $\mathbb{R}_{\geq 0}$, and the set $\{1, 2, \dots\}$ of positive integers as $\mathbb{N}_{>0}$.
- The set $\{1, \dots, n\}$, where $n \in \mathbb{N}_{>0}$, is compactly denoted as $[n]$; the empty set as $\emptyset$.
- Given a set S, we denote its convex hull with the symbol $\mathrm{co}(S)$.
- Vectors and matrices are marked in bold.
- Given a discrete set $S = \{s_1, \dots, s_n\}$, we denote as $\mathbb{R}^S$ (respectively, $\mathbb{R}^S_{\geq 0}$) the set of real (respectively, nonnegative real) $|S|$-dimensional vectors whose entries are denoted as $\boldsymbol{x}[s_1], \dots, \boldsymbol{x}[s_n]$. Given an element $s \in S$, we denote by $\boldsymbol{e}_s \in \mathbb{R}^S$ the canonical basis vector whose entries are all zeros except for $\boldsymbol{e}_s[s] = 1$.
- Similarly, given a discrete set S, we denote as $\mathbb{R}^{|S| \times |S|}$ (respectively, $\mathbb{R}^{|S| \times |S|}_{\geq 0}$) the set of real (respectively, nonnegative real) $|S| \times |S|$ square matrices $\boldsymbol{M}$ whose entries are denoted as $\boldsymbol{M}[s_r, s_c]$ ($s_r, s_c \in S$), where s_r corresponds to the row index and s_c to the column index.
- Given a discrete set S, we denote by Δ^S the simplex $\Delta^S := \{\boldsymbol{x} \in \mathbb{R}^S_{\geq 0} : \sum_{s \in S} \boldsymbol{x}[s] = 1\}$. The symbol Δ^n, with $n \in \mathbb{N}_{>0}$, is used to mean $\Delta^{[n]}$.
- Given a discrete set S, we use the symbol $\mathbb{S}^S \subseteq \mathbb{R}^{|S| \times |S|}_{\geq 0}$ to denote the set of stochastic matrices, that is, nonnegative square matrices whose columns all sum up to 1. The symbol $\mathbb{S}^n$, where $n \in \mathbb{N}_{>0}$, is used to mean $\mathbb{S}^{[n]}$.
- Given two functions $f : X \to Y$ and $g : Y \to Z$, we denote by $g \circ f : X \to Z$ their composition $\boldsymbol{x} \mapsto g(f(x))$.
- Given a set S and a function f, the *image of S via f* is denoted as $f(S) := \{f(s) : s \in S\}$.
- Given a proposition p, we denote with $\mathbb{1}_{\{p\}}$ the indicator function of that proposition: $\mathbb{1}_{\{p\}} = 1$ if p is true, and $\mathbb{1}_{\{p\}} = 0$ if not.
- Given a partially ordered set $(S, \prec)$ and two elements $s, s' \in S$, we use the standard derived symbols $s \preceq s'$ to mean that $(s = s') \vee (s \prec s')$, $s \succ s'$ to mean that $s' \prec s$, and $s \succeq s'$ to

mean that $s' \preceq s$. Furthermore, we use the crossed symbols $\not\prec, \not\preceq, \not\succ$, and $\not\succeq$ to mean that the relations $\prec, \preceq, \succ$, and $\succeq$ (respectively) do not hold.

- Several of the algorithms presented in this article take as input, give as output, or otherwise manipulate, linear functions. Therefore, to study the complexity of our routines, it is necessary to settle on a representation for such linear functions. Unless otherwise specified, we will always assume that a linear function f is stored in memory using coordinates relative to the canonical basis of their domain and codomain and call that representation the *canonical representation* of f, denoted $\langle f \rangle$. Specifically:

  – If f is a linear function from $\mathbb{R}^S$ (for some discrete set S) to $\mathbb{R}$, then its canonical representation $\langle f \rangle$ is the (unique) vector $\boldsymbol{v} \in \mathbb{R}^S$ such that

$$f(\boldsymbol{x}) = \boldsymbol{v}^\top \boldsymbol{x} \qquad \forall \, \boldsymbol{x} \in \mathbb{R}^S,$$

  where $\top$ denotes transposition.
  – If f is a linear function from $\mathbb{R}^S$ to $\mathbb{R}^S$ (for some discrete set S), then its canonical representation $\langle f \rangle$ is the (unique) matrix $\boldsymbol{M} \in \mathbb{R}^{|S| \times |S|}$ such that

$$f(\boldsymbol{x}) = \boldsymbol{M}\boldsymbol{x} \qquad \forall \, \boldsymbol{x} \in \mathbb{R}^S.$$

  – If F is a linear functional, mapping linear functions $\phi : \mathbb{R}^S \to \mathbb{R}^S$ to reals, then its canonical representation $\langle F \rangle$ is the (unique) matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{|S| \times |S|}$ such that

$$F(\phi) = \sum_{s_r, s_c \in S} \boldsymbol{\Lambda}[s_r, s_c] \cdot \langle \phi \rangle [s_r, s_c] \qquad \forall \, \phi : \mathbb{R}^S \to \mathbb{R}^S, \tag{1}$$

  where $\langle \phi \rangle$ is the canonical representation of ϕ.

2.2 Extensive-form Games

In this subsection, we introduce some standard concepts, terminology, and notation that we will use to deal with extensive-form games. A summary of the notation we introduce can be found in Table 1. Examples 2.1 and 2.5 demonstrate some of the notation in a simple extensive-form game.

An extensive-form game is played on an oriented rooted *game tree*. We denote by $\mathcal{H}$ the set of nodes of the game tree. Each node $h \in \mathcal{H}$ that is not a leaf of the game tree is called a *decision node* and has an associated player that acts at that node. In an n-player extensive-form game, the set of valid players is the set $[n] \cup \{c\}$, where c denotes the *chance player*—a fictitious player that selects actions according to a known fixed probability distribution and models exogenous stochasticity of the environment (for example, a roll of the dice or drawing a card from a deck). The player that acts at h is free to pick any one of the actions $\mathcal{A}(h)$ that are available at h. For each possible action $a \in \mathcal{A}(h)$, an edge connects h to the node to which the game transitions whenever action a is picked at h. Given a player $i \in [n] \cup \{c\}$, we denote with $\mathcal{H}^{(i)} \subseteq \mathcal{H}$ the set of all decision nodes that belong to Player i.

Leaves of the game tree are called *terminal nodes* and represent the outcomes of the game. As such, they are not associated with any acting player, and the set of actions is conventionally set to the empty set. The set of all terminal nodes in the game is denoted with the letter $\mathcal{Z}$. So, the set of all nodes in the game tree is the disjoint union $\mathcal{H} = \mathcal{H}^{(1)} \cup \cdots \cup \mathcal{H}^{(n)} \cup \mathcal{Z}$. When the game transitions to a terminal node $z \in \mathcal{Z}$, payoffs are assigned to each of the non-chance players by the set of functions $\{u^{(i)} : \mathcal{Z} \to \mathbb{R}\}_{i \in [n]}$. Furthermore, we let $p_c : \mathcal{Z} \to (0, 1)$ denote the function assigning each terminal node $z \in \mathcal{Z}$ to the product of probabilities of chance moves encountered on the path from the root of the game tree to z.

Table 1. Summary of Game-theoretic Notation Used in this Article

Symbol	Description
$\mathcal{H}$	Set of nodes of the game tree.
$\mathcal{H}^{(i)}$	Set of nodes at which Player i acts.
$\mathcal{A}(h)$	Actions available to the player acting at $h \in \mathcal{H}$ (empty set if h is a terminal node).
$\mathcal{J}^{(i)}$	Information partition of Player i.
$\mathcal{A}(j)$	Set of actions available at any node in the information set j.
$\mathcal{Z}$	Set of terminal nodes (leaves of the game tree).
$u^{(i)}(z)$	Payoff of Player i at terminal node $z \in \mathcal{Z}$.
$p_c(z)$	Product of probabilities of all the stochastic events on the path from the root to terminal node $z \in \mathcal{Z}$.
$\Sigma^{(i)}$	Set of sequences of Player i, defined as $\Sigma^{(i)} := \{(j, a) : j \in \mathcal{J}^{(i)}, a \in \mathcal{A}(j)\} \cup \{\varnothing\}$,
$\varnothing$	where the special element $\varnothing$ is called the *empty sequence*.
$\Sigma_*^{(i)}$	Set of sequences of Player i, excluding the empty sequence $\varnothing$.
$\sigma^{(i)}(z)$	Last sequence of Player i encountered on the path from the root to node $z \in \mathcal{Z}$.
$\sigma^{(i)}(j)$	Last sequence of Player i on the path from the root to any node in $j \in \mathcal{J}^{(i)}$.
$j \prec j'$	Information set $j \in \mathcal{J}^{(i)}$ is an ancestor of $j' \in \mathcal{J}^{(i)}$, that is, there exists a directed path in the game tree connecting a node $h \in j$ to some node $h' \in j'$.
$\sigma \prec \sigma'$	Sequence σ precedes sequence σ', where σ, σ' belong to the same player.
$\sigma \succeq j$	Sequence $\sigma = (j', a')$ is such that $j' \succeq j$.
$\Sigma_j^{(i)}$	Sequences at $j \in \mathcal{J}^{(i)}$ and all of its descendants, $\Sigma_j^{(i)} := \{\sigma \in \Sigma^{(i)} : \sigma \succeq j\}$.
$Q^{(i)}$	Sequence-form strategies of Player i (Definition 2.2).
$Q_j^{(i)}$	Sequence-form strategies for the subtree[2] rooted at $j \in \mathcal{J}^{(i)}$ (Definition 2.3).
$\Pi^{(i)}$	Deterministic sequence-form strategies of Player i.
$\Pi_j^{(i)}$	Deterministic sequence-form strategies for the subtree[2] rooted at $j \in \mathcal{J}^{(i)}$.
Π	Set of joint deterministic sequence-form strategies, $\Pi := \bigtimes_{i \in [n]} \Pi^{(i)}$.

2.2.1 Imperfect Information. To model imperfect information, the nodes $\mathcal{H}^{(i)}$ of each player $i \in [n]$ are partitioned into a collection $\mathcal{J}^{(i)}$ of set of nodes, called *information sets*. Each information set $j \in \mathcal{J}^{(i)}$ groups together nodes that Player i cannot distinguish between when he or she acts there. Since a player always knows what actions are available at a decision node, any two nodes h, h' belonging to the same information set j must have the same action set, that is, $\mathcal{A}(h) = \mathcal{A}(h')$ for all $h, h' \in j$. For that reason, we can safely overload notation and write $\mathcal{A}(j)$ to mean the set of actions available at any node that belongs to information set j.

As is standard in the literature, we assume that the extensive-form game has *perfect recall*, that is, information sets are such that no player forgets information once acquired. An immediate consequence of perfect recall is that, for any player $i \in [n]$ and any two nodes h, h' in the same information set $j \in \mathcal{J}^{(i)}$, the sequence of Player i's actions encountered along the path from the root to h and from the root to h' must coincide (or otherwise Player i would be able to distinguish among the nodes, since the player remembers all of the actions they played in the past). This

---

[2]The term "subtree" does not refer to a subtree of the game tree, but rather to a subtree of the partially ordered set $(\mathcal{J}^{(i)}, \prec)$. In other words, the term *subtree* here refers to the fact that the quantities are specified only at information set j and all of its descendants.

suggests the following partial ordering $\prec$ on the set $\mathcal{J}^{(i)}$: We write $j \prec j'$—and say that $j \in \mathcal{J}^{(i)}$ is an *ancestor* of $j' \in \mathcal{J}^{(i)}$ or equivalently that j' is a *descendant* of j—if there exist nodes $h' \in j'$ and $h \in j$ such that the path from the root of the game tree to h' passes through h.

It is a well-known consequence of perfect recall that the partially ordered set $(\mathcal{J}^{(i)}, \prec)$ is a forest for any player $i \in [n]$, in the precise sense that, given any information set $j \in \mathcal{J}^{(i)}$, the set of all of its predecessors forms a chain (that is, it is well-ordered by $\prec$).

2.2.2 Sequences. For any player $i \in [n]$, and given an information set $j \in \mathcal{J}^{(i)}$ and an action $a \in \mathcal{A}(j)$, we denote as $\sigma = (j, a)$ the sequence of Player i's actions encountered on the path from the root of the game tree down to action a (included) at any node in information set j. In perfect-recall extensive-form games, such a sequence is guaranteed to be uniquely determined, because paths that reach decision nodes belonging to the same information set identify the same sequence of Player i's actions. A special element $\varnothing$ denotes the *empty sequence* of Player i. Then, the set of Player i's sequences is defined as

$$\Sigma^{(i)} := \{(j, a) : j \in \mathcal{J}^{(i)}, a \in \mathcal{A}(j)\} \cup \{\varnothing\}.$$

Moreover, we let $\Sigma_*^{(i)} := \Sigma^{(i)} \setminus \{\varnothing\}$ be the set of all sequences of Player i other than the empty one.

Given a node $h \in \mathcal{H}$, we denote by $\sigma^{(i)}(j) \in \Sigma^{(i)}$ the last sequence (information set-action pair) of Player i encountered on the path from the root of the game tree to any node in j, also known as j's *parent sequence*. If Player i does not act before h, then $\sigma^{(i)}(h)$ is set to the empty sequence $\varnothing$. If $\sigma^{(i)}(j) = \varnothing$, then we say that information set j is a *root information set* of Player i, while, whenever $\sigma^{(i)}(j) = (j', a)$, we say that j' is the *immediate predecessor* of j, or equivalently that information set j is *immediately reachable* from sequence $\sigma^{(i)}(j)$. That nomenclature is supported by the observation that $j' \prec j$, and that Player i does not need to take other actions after choosing a at j' to reach j. Analogously, for all $z \in \mathcal{Z}$, we define $\sigma^{(i)}(z) \in \Sigma^{(i)}$ as the last sequence of Player i's actions encountered the path from the root of the game tree to terminal node z (notice that $\sigma^{(i)}(z) = \varnothing$ whenever Player i never plays on the path from the root to z).

Just like information sets, there exists a natural partial ordering on sequences, which we also denote with the same symbol $\prec$. For every $i \in [n]$ and any pair of sequences $\sigma, \sigma' \in \Sigma^{(i)}$, the relation $\sigma \prec \sigma'$ holds if $\sigma = \varnothing \neq \sigma'$, or if the sequences are of the form $\sigma = (j, a), \sigma' = (j', a')$, and the set of Player i's actions encountered on the path from the root of the tree to any node in j' includes playing action a at one of information set j's nodes. As for information sets, it is a direct consequence of the perfect recall assumption that the partially ordered set $(\Sigma^{(i)}, \prec)$ is a forest. Finally, we introduce the overloaded notation $\sigma \succeq j$ (or equivalently $j \preceq \sigma$), defined for any player $i \in [n]$, information set $j \in \mathcal{J}^{(i)}$, and sequence $\sigma \in \Sigma^{(i)}$, to mean that the sequence of Player i 's actions that is denoted by σ must lead the player to pass through (some node in) j; formally $\sigma = (j', a') \in \Sigma_*^{(i)} \wedge j' \succeq j$. With that, we let $\Sigma_j^{(i)} := \{\sigma \in \Sigma^{(i)} : \sigma \succeq j\} \subseteq \Sigma^{(i)}$ be the set of Player i's sequences that terminate at j or any of its descendant information sets.

Example 2.1. To illustrate some of the concepts and notation described so far, we consider the simple two-player extensive-form game in Figure 1, in which black round nodes belong to Player 1, and white round nodes belong to Player 2. The gray clusters of nodes identify the information sets. Since we chose different action numbers for different information sets, there exists a one-to-one correspondence between actions and sequences, and we will sometimes refer to sequences using the corresponding action number. For example, we will sometimes refer to sequence "3" to mean sequence $(B, 3)$, sequence "8" to mean sequence $(D, 8)$, and so on. Player 1 has four information sets—denoted A, B, C, and D—with two actions each. Player 2 only has two information sets, R and S, each with two actions. Information set D of Player 1 contains two nodes and models Player 1's
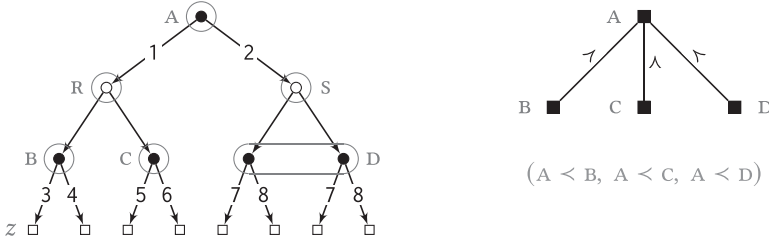
Fig. 1. (Left) Example of an extensive-form game with two players. Black round nodes belong to Player 1, white round nodes belong to Player 2. Small white square nodes represent terminal nodes. The gray partitions represent the information sets of the game. The numbers on the edges identify each of Player 1's actions. (Right): Forest of information sets of Player 1, corresponding to the partially ordered set $(\mathcal{J}^{(1)}, \prec)$.

lack of knowledge of the action taken by Player 2 at information set s. The partial ordering between information sets for Player 1 is $\text{A} \prec \text{B}, \text{A} \prec \text{C}, \text{A} \prec \text{D}$. Moreover, we have that $\sigma^{(1)}(\text{A}) = \varnothing$, $\sigma^{(1)}(\text{B}) = \sigma^{(1)}(\text{C}) = 1$, and $\sigma^{(1)}(\text{D}) = 2$. For the terminal node z in the picture, $\sigma^{(1)}(z) = 3$. Finally, we have that $\Sigma_{\text{A}}^{(1)} = \Sigma_{*}^{(1)} = \{1, 2, \ldots, 8\}$, $\Sigma_{\text{B}}^{(1)} = \{3, 4\}$, $\Sigma_{\text{C}}^{(1)} = \{5, 6\}$, and $\Sigma_{\text{D}}^{(1)} = \{7, 8\}$.

2.2.3 Sequence-form Strategies. Conceptually, a strategy for a player specifies a probability distribution over the actions at each information set for that player. So, perhaps the most intuitive representation of a strategy, called a *behavioral strategy* in the literature, is as a vector that assigns to each information set-action pair $(j, a) \in \Sigma_{*}^{(i)}$ the probability of picking action a at information set j. That representation has a major drawback: The probability of reaching any given terminal node $z \in \mathcal{Z}$ is expressed as the product of several entries in the vector (one per each action on the path from the root of the game tree to z), rendering critical quantities—including the expected utility of a player—a non-convex function of the behavioral strategies of the players. As is standard in the literature, to soundly overcome the issue of non-convexity, throughout this article, we will exclusively use a different representation of strategies, known as the *sequence-form* representation [30, 41, 47].

Like behavioral strategies, a *sequence-form strategy*[3] for Player $i \in [n]$ is a vector $\boldsymbol{q} \in \mathbb{R}_{\geq 0}^{\Sigma^{(i)}}$. However, unlike behavioral strategies, each entry $\boldsymbol{q}[(j, a)]$ of a sequence-form strategy $\boldsymbol{q}$ contains the *product* of the probabilities of playing all of Player i's actions on the path from the root of the game tree down to action a at information set j included. Furthermore, the entry $\boldsymbol{q}[\varnothing]$ corresponding to the empty sequence is defined as the constant value 1.

To ensure consistency, all sequence-form strategies must satisfy the probability-mass-conservation constraints

$$\boldsymbol{q}[\varnothing] = 1, \qquad \boldsymbol{q}[\sigma^{(i)}(j)] = \sum_{a \in \mathcal{A}(j)} \boldsymbol{q}[(j, a)], \quad \forall j \in \mathcal{J}^{(i)}.$$

The above probability-mass-conservation constraints are linear, and therefore the set of sequence-form strategies is a convex polytope, suggesting the following definition:

Definition 2.2. The *sequence-form strategy polytope* for Player $i \in [n]$ is the convex polytope

$$Q^{(i)} := \left\{ \boldsymbol{q} \in \mathbb{R}_{\geq 0}^{\Sigma^{(i)}} : \boldsymbol{q}[\varnothing] = 1 \quad \text{and} \quad \boldsymbol{q}[\sigma^{(i)}(j)] = \sum_{a \in \mathcal{A}(j)} \boldsymbol{q}[(j, a)], \ \forall j \in \mathcal{J}^{(i)} \right\}.$$

---

[3]Sequence-form strategies are also known under the term *realization plans* in the literature (e.g., Reference [47]). We will not use that latter term in this article.

As we mentioned in Section 2.2.2, the partially ordered set $(\mathcal{J}^{(i)}, \prec)$ is a forest. Thus, it makes sense to consider partial strategies that only specify behavior at an information set j and all of its descendants $j' \succ j$. We make that formal through the following definition:

Definition 2.3. Let $i \in [n]$ be a player and $j \in \mathcal{J}^{(i)}$ be an information set for Player i. The *set of sequence-form strategies for the subtree rooted at j*, denoted $\boldsymbol{Q}_j^{(i)}$, is the set of all vectors $\boldsymbol{q} \in \mathbb{R}_{\geq 0}^{\Sigma_j^{(i)}}$ such that probability-mass-conservation constraints hold at information set j and all of its descendants $j' \succ j$, specifically,

$$\boldsymbol{Q}_j^{(i)} := \left\{ \boldsymbol{q} \in \mathbb{R}_{\geq 0}^{\Sigma_j^{(i)}} : \sum_{a \in \mathcal{A}(j)} \boldsymbol{q}[(j, a)] = 1, \quad \text{and} \quad \boldsymbol{q}[\sigma^{(i)}(j')] = \sum_{a \in \mathcal{A}(j')} \boldsymbol{q}[(j', a)] \quad \forall j' \succ j \right\}. \quad (2)$$

2.2.4 Deterministic Sequence-form Strategies. Deterministic strategies are those that select, at each information set at which the player acts, exactly one action with probability one. Since the probability mass on each action is either 0 or 1, the set of deterministic sequence-form strategies for Player i—which we denote with the capital letter $\Pi^{(i)}$—corresponds exactly with the set of all sequence-form strategies whose components are all either 0 or 1.

Definition 2.4. The set of *deterministic sequence-form strategies* for Player $i \in [n]$ is the set

$$\Pi^{(i)} := \boldsymbol{Q}^{(i)} \cap \{0, 1\}^{\Sigma^{(i)}}.$$

Similarly, the set of *deterministic sequence-form strategies for the subtree[2] rooted at j* is

$$\Pi_j^{(i)} := \boldsymbol{Q}_j^{(i)} \cap \{0, 1\}^{\Sigma_j^{(i)}}.$$

The set of deterministic sequence-form strategies corresponds one-to-one to the game-theoretic notion of *reduced normal-form strategies* (e.g., von Stengel [47, Section 4]). Furthermore, Kuhn's Theorem [32] implies that

$$\boldsymbol{Q}^{(i)} = \text{co}(\Pi^{(i)}), \quad \boldsymbol{Q}_j^{(i)} = \text{co}(\Pi_j^{(i)}) \qquad \forall i \in [n], j \in \mathcal{J}^{(i)}.$$

When it is important to emphasize that an arbitrary sequence-form strategy $\boldsymbol{q} \in \boldsymbol{Q}^{(i)}$ (or $\boldsymbol{q} \in \boldsymbol{Q}_j^{(i)}$ for some $j \in \mathcal{J}^{(i)}$) of Player $i \in [n]$ need *not* be a deterministic sequence-form strategy, we will say that $\boldsymbol{q}$ is a *mixed* sequence-form strategy.

Given a sequence-form strategy $\boldsymbol{q} \in \boldsymbol{Q}^{(i)}$, it is possible to build an unbiased sampling scheme resulting in a (random) deterministic strategy $\boldsymbol{\pi} \in \Pi^{(i)}$ such that $\mathbb{E}[\boldsymbol{\pi}] = \boldsymbol{q}$. A natural unbiased sampling procedure is the following: Start from any root information set of Player i, that is, an information set $j \in \mathcal{J}^{(i)}$ such that $\sigma^{(i)}(j) = \varnothing$. Given any information set $j \in \mathcal{J}^{(i)}$, an action $a_j \in \mathcal{A}(j)$ is sampled with probability $\boldsymbol{q}[(j, a_j)]/\boldsymbol{q}[\sigma^{(i)}(j)]$; then, the same procedure is applied recursively to all information sets immediately reachable from sequence (j, a_j), that is, the information sets $j' \in \mathcal{J}^{(i)}$ such that $\sigma^{(i)}(j') = (j, a_j)$. The process is repeated for all the root information sets of Player i. The final deterministic sequence-form strategy $\boldsymbol{\pi}$ is obtained by setting $\boldsymbol{\pi}[(j, a_j)] = 1$ for each information set $j \in \mathcal{J}^{(i)}$ visited during the procedure, and all other entries equal to 0.

Finally, we denote as $\Pi := \times_{i \in [n]} \Pi^{(i)}$ the set of *joint* deterministic sequence-form strategies of all the players. Therefore, an element of Π is a tuple $\boldsymbol{\pi} = (\boldsymbol{\pi}^{(1)}, \ldots, \boldsymbol{\pi}^{(n)})$ specifying a deterministic sequence-form strategy $\boldsymbol{\pi}^{(i)}$ for each player $i \in [n]$.

Example 2.5. Continuing Example 2.1, in Figure 2, we provide one (mixed) sequence-form strategy $\boldsymbol{q} \in \boldsymbol{Q}^{(1)}$ and five deterministic sequence-form strategies $\{\boldsymbol{q}_{135}, \boldsymbol{q}_{136}, \boldsymbol{q}_{145}, \boldsymbol{q}_{27}, \boldsymbol{q}_{28}\} \subseteq \Pi^{(1)}$ for

q		q_{135}		q_{136}		q_{145}		q_{27}		q_{28}	
$\varnothing$	1.00	$\varnothing$	1.0	$\varnothing$	1.0	$\varnothing$	1.0	$\varnothing$	1.0	$\varnothing$	1.0
1	0.50	1	1.0	1	1.0	1	1.0	1	0.0	1	0.0
2	0.50	2	0.0	2	0.0	2	0.0	2	1.0	2	1.0
3	0.25	3	1.0	3	1.0	3	0.0	3	0.0	3	0.0
4	0.25	4	0.0	4	0.0	4	1.0	4	0.0	4	0.0
5	0.10	5	1.0	5	0.0	5	1.0	5	0.0	5	0.0
6	0.40	6	0.0	6	1.0	6	0.0	6	0.0	6	0.0
7	0.00	7	0.0	7	0.0	7	0.0	7	1.0	7	0.0
8	0.50	8	0.0	8	0.0	8	0.0	8	0.0	8	1.0

Fig. 2. Examples of sequence-form strategies for Player i in the game of Figure 1 (left).

the small game in Figure 1 (left). One can check that these vectors are valid sequence-form strategies by verifying that the probability-mass-conservation constraints of Definition 2.2 hold. Let us consider the mixed sequence-form strategy q. There, $q[(\text{A}, 1)] = q[(\text{A}, 2)] = 0.5$, and therefore Player 1 will select between actions 1 and 2 at information set A uniformly at random. Suppose Player 1 selects action 1. Then, if Player 1 reached information set B, then she would select actions 3 and 4 with probability $0.25/0.5 = 0.5$ each. However, if Player 1 reached information set C, then she would choose action 5 with probability $0.1/0.5 = 0.2$, and action 6 with probability $0.4/0.5 = 0.8$. Analogously, if Player 1 played action 2 at information set A, then upon reaching information set D she would play action 8 with probability $0.5/0.5 = 1$. In general, the probability of playing action a at a generic information set j can be obtained by dividing $q[(j, a)]$ by $q[\sigma^{(i)}(j)]$. As a second example, consider the deterministic sequence-form strategy q_{136}. When Player 1 plays according to that strategy, she will always choose action 1 at information set A, action 3 at information set B, and action 6 at information set C. It is impossible for the player to reach information set D given her strategy at A and correspondingly $q_{136}[7] = q_{136}[8] = 0$.

2.3 Extensive-form Correlated Equilibrium (EFCE)

Extensive-form correlated equilibrium has been proposed by von Stengel and Forges [48] as the natural counterpart to (normal-form) correlated equilibrium in extensive-form games. In an EFCE, before the beginning of the game the mediator draws a recommended action for each of the possible information sets that players may encounter in the game, according to some probability distribution defined over joint reduced normal-form strategies. These recommendations are not immediately revealed to each player. Instead, the mediator incrementally reveals relevant action recommendations as players reach new information sets. At any information set, the acting player is free to deviate from the recommended action, but doing so comes at the cost of future recommendations, which are no longer issued if the player deviates. In an EFCE, the recommended behavior is incentive-compatible for each player, that is, no player is strictly better off ever deviating from any of the mediator's recommended actions.

Before introducing the formal definition of EFCE, let us mention that one could directly extend the definition of CE [1] to extensive-form games, thereby obtaining what is usually called *normal-form correlated equilibrium* (NFCE) of the extensive-form game. In this case, the mediator draws and recommends a complete reduced normal-form strategy to each player before the game starts. Then, before the beginning of the game, each player can decide whether to follow

the recommended plan or deviate to an arbitrary strategy they desire. For arbitrary extensive-form games with perfect recall, the following inclusion of the set of equilibria holds: NFCE ⊆ EFCE.

Multiple equivalent definitions of EFCE can be given. In this article, we follow the equivalent formulation given by Farina et al. [15] based on the concept of *trigger agents* introduced by Gordon et al. [20] and Dudík and Gordon [13]. In what follows, we will assume that an extensive-form game has been fixed:

Definition 2.6 (Trigger Agent). Let $i \in [n]$ be a player, let $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$, and let $\hat{\pi} \in \Pi_j^{(i)}$. The $(\hat{\sigma}, \hat{\pi})$-*trigger agent* is the agent that plays the game as Player i according to the following rules:

- If the trigger agent has never been recommended to play action a at information set j, then the trigger agent will follow whatever recommendation is issued by the mediator.
- When the trigger agent reaches information set j and is recommended to play action a, we say that the trigger agent "gets triggered" by the *trigger sequence* $\hat{\sigma} = (j, a)$. This means that, from that point on, the trigger agent will disregard the recommendations and play according to the *continuation strategy* $\hat{\pi}$ from information set j onward (that is, at j and all of its descendant information sets).

An EFCE is a probability distribution $\mu \in \Delta^\Pi$ over Π such that for any player $i \in [n]$, trigger sequence $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$, and continuation strategy $\hat{\pi} \in \Pi_j^{(i)}$, the expected utility of the $(\hat{\sigma}, \hat{\pi})$-trigger agent is *not* strictly greater than the expected utility that Player i would obtain by always following all of the mediator's recommendations.

To turn the above condition into an analytic expression, it is useful to introduce the following additional quantities: Given a distribution $\mu \in \Delta^\Pi$, we let $r_\mu(z)$ be the probability that the game ends in terminal node $z \in \mathcal{Z}$ when all players follow recommendations issued by the mediator according to μ; in particular, for every $z \in \mathcal{Z}$, it holds:

$$r_\mu(z) := \sum_{\substack{(\pi^{(1)}, \ldots, \pi^{(n)}) \in \Pi \\ \pi^{(i)}[\sigma^{(i)}(z)]=1 \ \forall i \in [n]}} \mu[(\pi^{(1)}, \ldots, \pi^{(n)})],$$

where the summation is over all joint strategies $(\pi^{(1)}, \ldots, \pi^{(n)}) \in \Pi$ such that terminal node z is reachable when each player $i \in [n]$ plays according to $\pi^{(i)}$. Additionally, given a trigger sequence $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$ for a player $i \in [n]$ and a continuation strategy $\hat{\pi} \in \Pi_j^{(i)}$, we let $r_{\mu, \hat{\sigma} \to \hat{\pi}}^{(i)}(z)$ be the probability with which the $(\hat{\sigma}, \hat{\pi})$-trigger agent reaches terminal node z. In particular, for every terminal node $z \in \mathcal{Z}$ such that $\sigma^{(i)}(z) \succeq j$ it holds that:

$$r_{\mu, \hat{\sigma} \to \hat{\pi}}^{(i)}(z) := \left(\sum_{\substack{(\pi^{(1)}, \ldots, \pi^{(n)}) \in \Pi \\ \pi^{(i')}[\sigma^{(i')}(z)]=1 \ \forall i' \neq i \\ \pi^{(i)}[\hat{\sigma}]=1}} \mu[(\pi^{(1)}, \ldots, \pi^{(n)})] \right) \hat{\pi}[\sigma^{(i)}(z)].$$

We can now state the formal definition of EFCE and approximate EFCE.

Definition 2.7 (ϵ-EFCE; EFCE). Given $\epsilon \geq 0$, a probability distribution $\mu \in \Delta^\Pi$ is an ϵ-*approximate EFCE* (or ϵ-EFCE for short) if, for every player $i \in [n]$, trigger sequence $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$, and continuation strategy $\hat{\pi} \in \Pi_j^{(i)}$, the expected utility of the $(\hat{\sigma}, \hat{\pi})$-trigger agent is never larger than the expected utility that Player i would obtain by always following all of the mediator's

recommendations by more than the amount ϵ. In symbols,

$$\sum_{\substack{z \in Z \\ \sigma^{(i)}(z) \geq \hat{\sigma}}} u^{(i)}(z)\, p_c(z)\, r_\mu(z) \geq \sum_{\substack{z \in Z \\ \sigma^{(i)}(z) \geq j}} u^{(i)}(z)\, p_c(z)\, r^{(i)}_{\mu,\,\hat{\sigma} \to \hat{\pi}}(z) - \epsilon.$$

A probability distribution $\mu \in \Delta^\Pi$ is an EFCE if it is a 0-EFCE.

2.4 Regret Minimization and Phi-regret Minimization

In this article, we will make heavy use of a mathematical object—one of the core abstractions in the field of online optimization—called a *regret minimizer*.

Definition 2.8. Let $\mathcal{X}$ be a subset of an Euclidean space with suitable dimension. A *regret minimizer for $\mathcal{X}$* is an abstract model for a decision maker that repeatedly interacts with a black-box environment. At each time t, the regret minimizer interacts with the environment through two operations:

- NextElement has the effect that the regret minimizer will output an element $\boldsymbol{x}^t \in \mathcal{X}$;
- ObserveUtility(ℓ^t) provides the environment's feedback to the regret minimizer, in the form of a linear[4] utility function $\ell^t : \mathcal{X} \to \mathbb{R}$ that evaluates how good the last-output point $\boldsymbol{x}^t$ was. The utility function can depend adversarially on the outputs $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^{t-1}$ of the regret minimizer, but not on $\boldsymbol{x}^t$.[5]

Calls to NextElement and ObserveUtility keep alternating to each other: First, the regret minimizer will output a point $\boldsymbol{x}^1$, then it will received feedback ℓ^1 from the environment, then it will output a new point $\boldsymbol{x}^2$, and so on. The decision making encoded by the regret minimizer is *online*, in the sense that at each time t, the output of the regret minimizer can depend on the prior outputs $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^{t-1}$ and corresponding observed utility functions $\ell^1, \ldots, \ell^{t-1}$, but no information about future utilities is available. The objective for the regret minimizer is to output points so the *cumulative regret* (or simply *regret*)

$$R^T := \max_{\boldsymbol{x}^* \in \mathcal{X}} \sum_{t=1}^{T} \left(\ell^t(\boldsymbol{x}^*) - \ell^t(\boldsymbol{x}^t) \right) \tag{3}$$

grows asymptotically sublinearly in the time T. Many regret minimizers that guarantee a cumulative regret $R^T = O(\sqrt{T})$ at all times T for any convex and compact set $\mathcal{X}$ are known in the literature (see, e.g., Cesa-Bianchi and Lugosi [9]).

---

[4]Throughout the article, we use the term *linear* and *affine* when referring to a function f on a domain $\mathcal{X}$ to mean that f extends to a linear or affine function in the Euclidean space that contains $\mathcal{X}$.

[5]More precisely, throughout the article, we make the following, standard technical assumptions about the way randomness can be leveraged by the regret minimizer and by the environment it interacts with:

- At all t, the regret minimizer has access to a private source of randomness, which we model as a random vector with finite mean $\mathbf{S}^t$. Similarly, the environment has access to a private source of randomness, which we model again as a random vector with finite mean $\mathbf{E}^t$. All sources of randomness are independent, that is, $\{\mathbf{S}^1, \mathbf{E}^1, \mathbf{S}^2, \mathbf{E}^2, \ldots\}$ are independent random variables.
- At all t, the output $\boldsymbol{x}^t \in \mathcal{X}$ of the regret minimizer is a function of the past outputs $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^{t-1}$ and their corresponding feedbacks $\ell^1, \ldots, \ell^{t-1}$, as well as on the random outcome $\mathbf{S}^t$. So, $\boldsymbol{x}^t$ is measurable with respect to the σ-algebra generated by $\{\mathbf{E}^1, \ldots, \mathbf{E}^{t-1}, \mathbf{S}^1, \ldots, \mathbf{S}^t\}$.
- At all t, the linear utility function ℓ^t constructed by the environment is a function of the past outputs $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^{t-1}$ and their corresponding feedback $\ell^1, \ldots, \ell^{t-1}$, as well as of the random outcome $\mathbf{E}^t$. So, ℓ^t is measurable with respect to the σ-algebra generated by $\{\mathbf{E}^1, \ldots, \mathbf{E}^t, \mathbf{S}^1, \ldots, \mathbf{S}^{t-1}\}$.
- Consequently, at all t, ℓ^t is conditionally independent of $\boldsymbol{x}^t$, given the σ-algebra $\mathcal{F}^{t-1} := \sigma(\mathbf{E}^1, \ldots, \mathbf{E}^{t-1}, \mathbf{S}^1, \ldots, \mathbf{S}^{t-1})$ generated by all past random outcomes.

A *phi-regret minimizer* is an extension of the concept of a regret minimizer introduced by Stoltz and Lugosi [44], building on previous work by Greenwald and Jafari [21].

Definition 2.9. Given a set $\mathcal{X}$ of points and a set Φ of affine transformations $\phi : \mathcal{X} \to \mathcal{X}$, a *phi-regret minimizer relative to Φ for the set $\mathcal{X}$*—abbreviated in the term "*Φ-regret minimizer*"—is an object with the same semantics and operations of a regret minimizer, but whose quality metric is its *cumulative phi-regret relative to Φ* (or simply *phi-regret relative to Φ*, or *Φ-regret* for short)

$$R^T := \max_{\phi^* \in \Phi} \sum_{t=1}^{T} \left(\ell^t(\phi^*(\mathbf{x}^t)) - \ell^t(\mathbf{x}^t) \right), \tag{4}$$

instead of the regular cumulative regret defined in (3). Once again, the goal for a phi-regret minimizer is to guarantee that its phi-regret grows asymptotically sublinearly as time T increases.

In the special case of the set of constant transformations $\Phi^{\text{const}} := \{ \mathcal{X} \ni \mathbf{x} \mapsto \hat{\mathbf{x}} : \hat{\mathbf{x}} \in \mathcal{X} \}$, the definition of cumulative phi-regret (4) reduces to that of cumulative regret given in (3). So, a regret minimizer is a special case of a phi-regret minimizer.

A general construction by Gordon et al. [20] gives a way to construct a Φ-regret minimizer for $\mathcal{X}$ starting from any regret minimizer (in the sense of Definition 2.8) for the convex hull co(Φ) of the set of functions Φ. Specifically, let $\mathcal{R}_\Phi$ be a deterministic[6] regret minimizer for the set of transformations co(Φ) whose cumulative regret grows sublinearly and assume that every $\phi \in$ co(Φ) admits a fixed point $\phi(\mathbf{x}) = \mathbf{x} \in \mathcal{X}$. Then, a Φ-regret minimizer $\mathcal{R}$ can be constructed starting from $\mathcal{R}_\Phi$ as follows:

- Each call to $\mathcal{R}$.NextElement first calls NextElement on $\mathcal{R}_\Phi$ to obtain the next transformation $\phi^t \in$ co(Φ). Then, a fixed point $\mathbf{x}^t = \phi^t(\mathbf{x}^t)$ is computed and output.
- Each call to $\mathcal{R}$.ObserveUtility(ℓ^t) with linear utility function ℓ^t constructs the linear utility function $L^t : \phi \mapsto \ell^t(\phi(\mathbf{x}^t))$, where $\mathbf{x}^t$ is the last-output strategy, and passes it to $\mathcal{R}_\Phi$ by calling $\mathcal{R}_\Phi$.ObserveUtility(L^t).[7]

The proof of correctness of the above construction is deceptively simple, and we recall it next. Since $\mathcal{R}_\Phi$ outputs transformations $\phi^1, \phi^2, \dots$ and receives utilities $\phi \mapsto \ell^1(\phi(\mathbf{x}^1)), \phi \mapsto \ell^2(\phi(\mathbf{x}^2)), \dots$, its cumulative regret R_Φ^T is

$$R_\Phi^T = \max_{\phi^* \in \text{co}(\Phi)} \sum_{t=1}^{T} \left(\ell^t(\phi^*(\mathbf{x}^t)) - \ell^t(\phi^t(\mathbf{x}^t)) \right).$$

Hence, since $\mathbf{x}^t = \phi^t(\mathbf{x}^t)$ is a fixed point of ϕ^t, we can write

$$R_\Phi^T = \max_{\phi^* \in \text{co}(\Phi)} \sum_{t=1}^{T} \left(\ell^t(\phi^*(\mathbf{x}^t)) - \ell^t(\mathbf{x}^t) \right) \geq \max_{\phi^* \in \Phi} \sum_{t=1}^{T} \left(\ell^t(\phi^*(\mathbf{x}^t)) - \ell^t(\mathbf{x}^t) \right), \tag{5}$$

where the inequality follows from the observation that co(Φ) $\supseteq \Phi$. The right-hand side is exactly the cumulative Φ-regret R^T incurred by $\mathcal{R}$, as defined in (4). So, because the regret cumulated by $\mathcal{R}_\Phi$ grows sublinearly by hypothesis, then so does the Φ-regret cumulated by $\mathcal{R}$.

---

[6]The assumption that $\mathcal{R}_\Phi$ is deterministic immediately guarantees that, at all times t, *any* linear utility function given as feedback to $\mathcal{R}_\Phi$ is conditionally independent of the last output ϕ^t, given the random outcomes used by the environments $\mathbf{E}^1, \dots, \mathbf{E}^{t-1}$ (cf. Footnote 5).

[7]On the surface, it might look like L^t is independent of the last output ϕ^t of the regret minimizer $\mathcal{R}_\Phi$, and thus, that it trivially satisfies the requirements of Definition 2.8. However, that is not true: $\mathbf{x}^t$ is a fixed point of ϕ^t, and, since $\mathbf{x}^t$ enters into the definition of L^t, if $\mathcal{R}_\Phi$ picks ϕ^t randomly, then it might very well be that L^t is not conditionally independent of ϕ^t. We sidestep this issue by requiring that $\mathcal{R}_\Phi$ is deterministic (cf. Footnote 6).

3 TRIGGER REGRET AND RELATIONSHIP WITH EFCE

In this section, we formalize a notion of *trigger deviation function*, building on an idea by Gordon et al. [20, Section 3]. We also introduce a connected notion of *trigger regret minimization*, which is an instance of phi-regret minimization, as recalled in Section 2.4. The central result of this section, Theorem 3.8, establishes a formal connection between EFCE and agents that minimize their trigger regret, thereby extending and generalizing the classic connection between correlated equilibrium and no-internal-regret in normal-form games [22] to the extensive-form game counterpart.

Definition 3.1 (Trigger Deviation Function). Let $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$, and $\hat{\pi} \in \Pi_j^{(i)}$. We call "*trigger deviation function corresponding to trigger $\hat{\sigma}$ and continuation strategy $\hat{\pi}$*" any linear function $f : \mathbb{R}^{\Sigma^{(i)}} \to \mathbb{R}^{\Sigma^{(i)}}$ whose effect on deterministic sequence-form strategies is as follows:

- all strategies $\pi \in \Pi^{(i)}$ that do not prescribe the sequence $\hat{\sigma}$ are left unmodified. In symbols,

$$f(\pi) = \pi \qquad \forall \, \pi \in \Pi^{(i)} : \pi[\hat{\sigma}] = 0; \tag{6}$$

- all strategies $\pi \in \Pi^{(i)}$ that prescribe sequence $\hat{\sigma} = (j, a)$ are modified so the behavior at j and all of its descendants is replaced with the behavior prescribed by the continuation strategy $\hat{\pi}$. In symbols,

$$f(\pi)[\sigma] = \begin{cases} \pi[\sigma] & \text{if } \sigma \not\succeq j \\ \hat{\pi}[\sigma] & \text{if } \sigma \succeq j, \end{cases} \qquad \forall \, \sigma \in \Sigma^{(i)}, \pi \in \Pi^{(i)} : \pi[\hat{\sigma}] = 1. \tag{7}$$

Trigger deviation functions are simpler than the *extensive-form transformations* described by Gordon et al. [20]. Specifically, extensive-form transformations allow one to specify more than one trigger sequence (together with different continuation strategies, one for each specified trigger sequence), whereas our notion of trigger deviation functions only contemplates a single trigger sequence. Consequently, the set of all trigger deviation functions is significantly smaller, and simpler, than the set of all extensive-form transformations. The simpler structural properties of the set of trigger deviation functions, explored in Section 4.3, will enable us to construct an efficient regret minimizer for the convex hull of the set of all trigger deviation functions.

At this stage, it is technically unclear whether a linear function that satisfies Definition 3.1 exists for all valid choices of $\hat{\sigma}$ and $\hat{\pi}$. We show that this is indeed the case by explicitly exhibiting a linear function, which we call the *canonical trigger deviation function*. We start with a definition:

Definition 3.2. Let $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$ and $\mathbf{y} \in \mathbb{R}_{\geq 0}^{\Sigma_j^{(i)}}$. We denote with $M_{\hat{\sigma} \to \mathbf{y}}^{(i)} \in \mathbb{R}_{\geq 0}^{|\Sigma^{(i)}| \times |\Sigma^{(i)}|}$ the matrix whose entries are defined as

$$M_{\hat{\sigma} \to \mathbf{y}}^{(i)}[\sigma_r, \sigma_c] = \begin{cases} 1 & \text{if } \sigma_c \not\succeq \hat{\sigma} \text{ and } \sigma_r = \sigma_c \\ \mathbf{y}[\sigma_r] & \text{if } \sigma_c = \hat{\sigma} \text{ and } \sigma_r \succeq j \\ 0 & \text{otherwise,} \end{cases} \qquad \forall \, \sigma_r, \sigma_c \in \Sigma^{(i)}. \tag{8}$$

Furthermore, we denote with the symbol $\phi_{\hat{\sigma} \to \mathbf{y}}^{(i)}$ the linear map $\mathbb{R}^{\Sigma^{(i)}} \ni \mathbf{x} \mapsto M_{\hat{\sigma} \to \mathbf{y}}^{(i)} \mathbf{x}$.

Intuitively, by recalling that the columns of a matrix are the images of the unit basis vectors under the linear map, we observe that $M_{\hat{\sigma} \to \mathbf{y}}^{(i)}$ maps the canonical basis vector $e_{\hat{\sigma}}$ to $\mathbf{y}$, the vector e_σ is mapped to itself for each $\sigma \not\succeq \hat{\sigma}$, and e_σ is mapped to zero for every $\sigma \succ \hat{\sigma}$.

In the following, we will focus on trigger deviation functions defined through the linear mapping of Equation (8). We call such deviation functions *canonical trigger deviation functions*.

Definition 3.3. Let $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$ and $\hat{\boldsymbol{\pi}} \in \Pi_j^{(i)}$. The function $\phi_{\hat{\sigma} \to \hat{\boldsymbol{\pi}}}^{(i)}$ is called the "*canonical trigger deviation function corresponding to trigger $\hat{\sigma}$ and continuation strategy $\hat{\boldsymbol{\pi}}$.*" Furthermore, the set of all canonical trigger deviation functions is denoted with the symbol

$$\Psi^{(i)} := \left\{ \phi_{\hat{\sigma} \to \hat{\boldsymbol{\pi}}}^{(i)} : \hat{\sigma} = (j, a) \in \Sigma_*^{(i)}, \hat{\boldsymbol{\pi}} \in \Pi_j^{(i)} \right\}.$$

LEMMA 3.4. *For any $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$ and $\hat{\boldsymbol{\pi}} \in \Pi_j^{(i)}$, the linear function $\phi_{\hat{\sigma} \to \hat{\boldsymbol{\pi}}}^{(i)}$ as defined in Definition 3.3 is a trigger deviation function in the sense of Definition 3.1.*

PROOF. The proof just amounts to a simple application of several definitions. Let $\boldsymbol{\pi} \in \Pi^{(i)}$ be an arbitrary deterministic sequence-form strategy. By expanding the matrix-vector multiplication $\boldsymbol{M}_{\hat{\sigma} \to \hat{\boldsymbol{\pi}}}^{(i)} \boldsymbol{\pi}$ using the Definition (8), we obtain that for all $\sigma \in \Sigma^{(i)}$

$$(\boldsymbol{M}_{\hat{\sigma} \to \hat{\boldsymbol{\pi}}}^{(i)} \boldsymbol{\pi})[\sigma] = \boldsymbol{\pi}[\sigma] \mathbb{1}_{\{\sigma \not\geq \hat{\sigma}\}} + \hat{\boldsymbol{\pi}}[\sigma] \boldsymbol{\pi}[\hat{\sigma}] \mathbb{1}_{\{\sigma \geq j\}}. \tag{9}$$

There are only two possibilities:

- If $\boldsymbol{\pi}[\hat{\sigma}] = 0$, then (9) simplifies to

$$(\boldsymbol{M}_{\hat{\sigma} \to \hat{\boldsymbol{\pi}}}^{(i)} \boldsymbol{\pi})[\sigma] = \begin{cases} \boldsymbol{\pi}[\sigma] & \text{if } \sigma \not\geq \hat{\sigma} \\ 0 & \text{otherwise.} \end{cases}$$

  Since by case hypothesis the probability of the sequence of actions from the root of the game tree down to $\hat{\sigma}$ is zero, then necessarily the probability of any longer sequence of actions $\sigma \geq \hat{\sigma}$ must be zero as well, that is, $\boldsymbol{\pi}[\sigma] = 0$ for all $\sigma \geq \hat{\sigma}$. So, $\boldsymbol{M}_{\hat{\sigma} \to \hat{\boldsymbol{\pi}}}^{(i)} \boldsymbol{\pi} = \boldsymbol{\pi}$ and (6) holds.

- Conversely, assume $\boldsymbol{\pi}[\hat{\sigma}] = 1$. This means that at information set $j \in \mathcal{J}^{(i)}$ action a is selected (with probability 1), and therefore $\boldsymbol{\pi}[\sigma] = 0$ for all $\sigma = (j, a') : a' \in A_j, a' \neq a$. This means that $\boldsymbol{\pi}[\sigma] \mathbb{1}_{\{\sigma \not\geq \hat{\sigma}\}} = \boldsymbol{\pi}[\sigma] \mathbb{1}_{\{\sigma \not\geq j\}}$ for all $\sigma \in \Sigma^{(i)}$. Substituting that equality into (9) gives Equation (7), as we wanted to show. □

Since $\Pi_j^{(i)} \subseteq \{0, 1\}^{\Sigma_j^{(i)}}$, and $|\Sigma_j^{(i)}| \leq |\Sigma^{(i)}|$ by definition, we have the following immediate bound on the number of trigger deviation functions:

LEMMA 3.5. *The number $|\Psi^{(i)}|$ of canonical trigger deviation functions for any player $i \in [n]$ is upper bounded by $|\Sigma^{(i)}| \cdot 2^{|\Sigma^{(i)}|}$.*

In the following example, we show three examples of canonical trigger deviation functions operate. In particular, we show how they modify some deterministic sequence-form strategies on a simple extensive-form game.

Example 3.6. We build on the small extensive-form game of Figure 1, and the sequence-form strategies defined in Example 2.1, to provide some concrete intuition behind canonical trigger deviation functions as defined in Definition 3.3.

- First, let us consider the trigger deviation function $\phi_a := \phi_{(A,1) \to \hat{\boldsymbol{\pi}}_a}^{(1)}$, where the trigger sequence is $\hat{\sigma} = (A, 1)$, and the continuation strategy $\hat{\boldsymbol{\pi}}_a$ is such that Player 1 plays action 2 at information set A, and subsequently sequence 7 at information set D. The matrix $\boldsymbol{M}_a := \boldsymbol{M}_{(A,1) \to \hat{\boldsymbol{\pi}}_a}^{(1)}$ corresponding to ϕ_a is reported in Figure 3 (left). To illustrate the effect of this linear mapping on sequence-form strategies, we provide some examples using the deterministic sequence-form strategy vectors defined in Figure 2. First, we observe that any deterministic sequence-form strategy choosing action 1 with probability 1 triggers a deviation that follows the continuation strategy $\hat{\boldsymbol{\pi}}$. The deviation for those sequence-form strategies results in a final deterministic sequence-form strategy equal to $\boldsymbol{q}_{27}$. For example, using

M_a	M_b	M_c
Trigger sequence: (A,1)	Trigger sequence: (A,2)	Trigger sequence: (B,3)
Continuation: 2, 7	Continuation: 1, 3, 5	Continuation: 4

M_a, columns $\varnothing\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8$, rows $\varnothing,1,\dots,8$:

$$
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

M_b:

$$
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

M_c:

$$
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

Fig. 3. Matrices defining different canonical trigger deviation functions (Definition 3.3) for the simple extensive-form game of Figure 1. Entries highlighted in dark gray represent the entries of the matrix defined in the second case of Equation (8). Let $\hat{\sigma} = (j, a) \in \Sigma_*^{(1)}$ be the trigger sequence of the trigger deviation function. All indices (σ_r, σ_c) such that $\sigma_r, \sigma_c \geq j$ are highlighted in light gray.

some of the deterministic-sequence form strategies of Figure 2, it can be easily verified (by working out the matrix-vector product) that:

$$M_a q_{135} = M_a q_{136} = M_a q_{145} = q_{27}.$$

However, deterministic sequence-form strategies that do not select sequence 1 are left unmodified by the linear mapping. For instance,

$$M_a q_{28} = q_{28} \quad \text{and} \quad M_a q_{27} = q_{27}.$$

- Second, we examine the trigger deviation function $\phi_b := \phi_{(A,2)\to\hat{\pi}_b}^{(1)}$ for trigger sequence $\hat{\sigma} = (A, 2)$, where the continuation strategy $\hat{\pi}_b$ is defined so Player 1 plays action 1 at information set A, sequence 3 at information set B, and action 5 at information set C. The corresponding matrix $M_b := M_{(A,2)\to\hat{\pi}_b}^{(1)}$ is reported in Figure 3 (middle). As in the previous case, all deterministic sequence-form strategy vectors that put probability 1 on action 2 at information set A are modified so that the strategy at A and its descendants B,C,D matches the continuation strategy. For example, we have that

$$M_b q_{27} = M_b q_{28} = q_{135}.$$

Furthermore, sequence-form strategies that do not put probability 1 on sequence (A, 2) are left unchanged. So, for example,

$$M_b q_{136} = q_{136} \quad \text{and} \quad M_b q_{145} = q_{145}.$$

- As a final example, Figure 3 (right) reports the deviation matrix M_c corresponding to a trigger deviation function $\phi_c := \phi_{(B,3)\to\hat{\pi}_c}^{(1)}$, corresponding to trigger sequence $\hat{\sigma} = (B, 3)$ and continuation strategy $\hat{\pi}_c$ selecting action 4 at information set B. Here, we have that $M_c q_{135} = M_c q_{145}$, and $M_c q_{145} = q_{145}$.

We are now ready to define the concept of trigger regret minimization, which extends and generalizes the homonymous notion in the conference version of this article [8], as well as the notion of internal regret minimization in normal-form games.

Definition 3.7. For every $i \in [n]$, we call *trigger regret minimizer for player i* any $\Psi^{(i)}$-regret minimizer for the set of deterministic sequence-form strategies $\Pi^{(i)}$.

The following theorem shows that if each player $i \in [n]$ in the game plays according to a $\Psi^{(i)}$-regret minimizer, then the empirical frequency of play approaches the set of EFCEs. The result is proved starting from the definition of cumulative $\Psi^{(i)}$-regret, which is manipulated by using the definition of trigger deviation functions. Then, we rewrite the empirical frequency of play, replacing the summation over joint deterministic sequence-form strategies played at each $t \in [T]$ with a weighted sum depending on the empirical frequency of play (see Equation (16)), which allows us to recover the definition of EFCE (Definition 2.7). We also remark that, to define the linear utility functions in the following theorem, we assume to be working under the full-information feedback model: The regret minimizer for Player i has access to the utility function $u^{(i)}$ of Player i, the function p_c encoding probabilities of chance moves, and the pure sequence-form strategies $\pi^{(i'), t}$ for each $i' \neq i$. Extending the result to the bandit-information feedback setting is left as an open question.

THEOREM 3.8. *For each player $i \in [n]$, let $\boldsymbol{\pi}^{(i), 1}, \boldsymbol{\pi}^{(i), 2}, \dots, \boldsymbol{\pi}^{(i), T} \in \Pi^{(i)}$ be deterministic sequence-form strategies whose cumulative $\Psi^{(i)}$-regret with respect to the sequence of linear utility functions*

$$\ell^{(i), t} : \Pi^{(i)} \ni \boldsymbol{\pi}^{(i)} \mapsto \sum_{z \in Z} u^{(i)}(z) \, p_c(z) \left(\prod_{i' \neq i} \boldsymbol{\pi}^{(i'), t}[\sigma^{(i')}(z)] \right) \boldsymbol{\pi}^{(i)}[\sigma^{(i)}(z)] \tag{10}$$

is $R^{(i), T}$. Then, the empirical frequency of play defined as the probability distribution $\boldsymbol{\mu} \in \Delta^{\Pi}$ that draws each joint profile $(\boldsymbol{\pi}^{(1)}, \dots, \boldsymbol{\pi}^{(n)}) \in \Pi$ with probability

$$\boldsymbol{\mu}[(\boldsymbol{\pi}^{(1)}, \dots, \boldsymbol{\pi}^{(n)})] \coloneqq \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_{\{(\boldsymbol{\pi}^{(1), t}, \dots, \boldsymbol{\pi}^{(n), t}) = (\boldsymbol{\pi}^{(1)}, \dots, \boldsymbol{\pi}^{(n)})\}}$$

is an ϵ-EFCE, where $\epsilon \coloneqq \frac{1}{T} \max_{i \in [n]} R^{(i), T}$.

PROOF. It is immediate to check that $\boldsymbol{\mu}$ is indeed a valid element of the $|\Pi|$-simplex. Furthermore, the utility function $\ell^{(i), t}$ clearly satisfies the requirement of being independent of $\boldsymbol{\pi}^{(i), t}$, for all $i \in [n]$. We will show that $\boldsymbol{\mu}$ defines an ϵ-EFCE by verifying that the definition holds (Definition 2.7). Fix any player $i \in [n]$, trigger sequence $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$, and continuation strategy $\hat{\boldsymbol{\pi}} \in \Pi_j^{(i)}$. Since by hypothesis the cumulative $\Psi^{(i)}$-regret is upper bounded by $R^{(i), T}$, and $R^{(i), T} \leq T\epsilon$ by definition of ϵ, we must have

$$T\epsilon \geq \sum_{t=1}^{T} \ell^{(i), t}\left(\phi_{\hat{\sigma} \to \hat{\pi}}^{(i)}(\boldsymbol{\pi}^{(i), t}) \right) - \ell^{(i), t}\left(\boldsymbol{\pi}^{(i), t} \right).$$

By expanding the definition of the utility function, which was given in (10), the previous inequality is equivalent to

$$T\epsilon \geq \sum_{t=1}^{T} \sum_{z \in \mathcal{Z}} \alpha_z^{(i), t} \cdot \left(\phi_{\hat{\sigma} \to \hat{\pi}}^{(i)}(\boldsymbol{\pi}^{(i), t})[\sigma^{(i)}(z)] - \boldsymbol{\pi}^{(i), t}[\sigma^{(i)}(z)] \right), \tag{11}$$

where we used the symbol

$$\alpha_z^{(i),\,t} := u^{(i)}(z)\,p_c(z)\left(\prod_{i'\neq i}\pi^{(i'),\,t}[\sigma^{(i')}(z)]\right)$$

to lighten the notation. Since $\phi_{\hat\sigma\to\hat\pi}^{(i)}$ is a trigger deviation function (Lemmas 3.4), we have that (6) and (7) hold, and in particular it follows that

$$\phi_{\hat\sigma\to\hat\pi}^{(i)}(\pi^{(i),\,t})[\sigma] = \pi^{(i),\,t}[\sigma]$$

for all $t = 1, \ldots, T$ and $\sigma \not\trianglerighteq j$. So, the summation term in (11) is zero for all terminal states $z \in \mathcal{Z}$ such that $\sigma^{(i)}(z) \not\trianglerighteq j$, and thus we can safely restrict the domain of the summation over terminal states $z \in \mathcal{Z}_j^{(i)} := \{z \in \mathcal{Z} : \sigma^{(i)}(z) \trianglerighteq j\}$ only, obtaining

$$T\epsilon \geq \sum_{t=1}^{T}\sum_{z\in\mathcal{Z}_j^{(i)}}\alpha_z^{(i),\,t}\cdot\left(\phi_{\hat\sigma\to\hat\pi}^{(i)}(\pi^{(i),\,t})[\sigma^{(i)}(z)] - \pi^{(i),\,t}[\sigma^{(i)}(z)]\right). \tag{12}$$

We now study the term $\phi_{\hat\sigma\to\hat\pi}^{(i)}(\pi^{(i),\,t})[\sigma^{(i)}(z)]$ for a generic $t \in \{1, \ldots, T\}$ and $z \in \mathcal{Z}_j^{(i)}$ by splitting into cases contingent on the value of $\pi^{(i),\,t}[\hat\sigma] \in \{0, 1\}$. If $\pi^{(i),\,t}[\hat\sigma] = 0$, then (6) applies, and therefore

$$\phi_{\hat\sigma\to\hat\pi}^{(i)}(\pi^{(i),\,t})[\sigma^{(i)}(z)] - \pi^{(i),\,t}[\sigma^{(i)}(z)] = 0.$$

If, on the contrary, $\pi^{(i),\,t}[\hat\sigma] = 1$, then (7) applies, and $\phi_{\hat\sigma\to\hat\pi}^{(i)}(\pi^{(i),\,t})[\sigma^{(i)}(z)] = \hat\pi[\sigma^{(i)}(z)]$, where we used the fact that $\sigma^{(i)}(z) \trianglerighteq j$ by definition of $z \in \mathcal{Z}_j^{(i)}$. So, at all $t = 1, \ldots, T$ and for all $z \in \mathcal{Z}_j^{(i)}$, it holds that

$$\phi_{\hat\sigma\to\hat\pi}^{(i)}(\pi^{(i),\,t})[\sigma^{(i)}(z)] - \pi^{(i),\,t}[\sigma^{(i)}(z)] = \pi^{(i),\,t}[\hat\sigma]\left(\hat\pi[\sigma^{(i)}(z)] - \pi^{(i),\,t}[\sigma^{(i)}(z)]\right),$$

and thus (12) can be equivalently written as

$$T\epsilon \geq \sum_{t=1}^{T}\sum_{z\in\mathcal{Z}_j^{(i)}}\pi^{(i),\,t}[\hat\sigma]\,\alpha_z^{(i),\,t}\cdot\left(\hat\pi[\sigma^{(i)}(z)] - \pi^{(i),\,t}[\sigma^{(i)}(z)]\right). \tag{13}$$

We now make the crucial observation that time t appears in $\alpha_z^{(i),\,t}$ and (13) only as a superscript in the strategies $\pi^{(1),\,t}, \ldots, \pi^{(n),\,t}$, and nowhere else. Therefore, by introducing the functions

$$\alpha_z^{(i)} : \Pi \ni (\pi^{(1)}, \ldots, \pi^{(n)}) \mapsto u^{(i)}(z)\,p_c(z)\left(\prod_{i'\neq i}\pi^{(i')}[\sigma^{(i')}(z)]\right),\text{ and} \tag{14}$$

$$v_{\hat\sigma\to\hat\pi}^{(i)} : \Pi \ni \pi = (\pi^{(1)}, \ldots, \pi^{(n)}) \mapsto \sum_{z\in\mathcal{Z}_j^{(i)}}\pi^{(i)}[\hat\sigma]\,\alpha_z^{(i)}(\pi)\cdot\left(\hat\pi[\sigma^{(i)}(z)] - \pi^{(i)}[\sigma^{(i)}(z)]\right), \tag{15}$$

we can rewrite (13) as

$$T\epsilon \geq \sum_{t=1}^{T}v_{\hat\sigma\to\hat\pi}^{(i)}(\pi^{(1),\,t}, \ldots, \pi^{(n),\,t}) = \sum_{t=1}^{T}\sum_{\pi\in\Pi}\mathbb{1}_{\{(\pi^{(1),\,t},\ldots,\pi^{(n),\,t})=\pi\}}\cdot v_{\hat\sigma\to\hat\pi}^{(i)}(\pi)$$

$$= \sum_{\pi\in\Pi}\left(\sum_{t=1}^{T}\mathbb{1}_{\{(\pi^{(1),\,t},\ldots,\pi^{(n),\,t})=\pi\}}\right)v_{\hat\sigma\to\hat\pi}^{(i)}(\pi) = T\sum_{\pi\in\Pi}\mu[\pi]v_{\hat\sigma\to\hat\pi}^{(i)}(\pi), \tag{16}$$

where we used the definition of μ in the third equality. Dividing by T in (16), we can further write

$$\epsilon \geq \sum_{\pi \in \Pi} \mu[\pi]\, v^{(i)}_{\hat{\sigma} \to \hat{\pi}}(\pi). \tag{17}$$

Finally, by expanding the definition of $v^{(i)}_{\hat{\sigma} \to \hat{\pi}}$ in (17),

$$\epsilon \geq \sum_{\pi = (\pi^{(1)}, \ldots, \pi^{(n)}) \in \Pi} \left(\mu[\pi] \sum_{z \in \mathcal{Z}^{(i)}_j} \pi^{(i)}[\hat{\sigma}]\, \alpha^{(i)}_z(\pi) \cdot \left(\hat{\pi}[\sigma^{(i)}(z)] - \pi^{(i)}[\sigma^{(i)}(z)] \right) \right)$$

$$= \sum_{z \in \mathcal{Z}^{(i)}_j} \sum_{\pi = (\pi^{(1)}, \ldots, \pi^{(n)}) \in \Pi} \mu[\pi]\, \pi^{(i)}[\hat{\sigma}]\, \alpha^{(i)}_z(\pi) \cdot \left(\hat{\pi}[\sigma^{(i)}(z)] - \pi^{(i)}[\sigma^{(i)}(z)] \right). \tag{18}$$

The right-hand side of (18) can be simplified further by noticing that, by definition of $\alpha^{(i)}_z(\pi)$,

$$\pi^{(i)}[\hat{\sigma}]\alpha^{(i)}_z(\pi^{(1)}, \ldots, \pi^{(n)}) = u^{(i)}(z) p_c(z) \pi^{(i)}[\hat{\sigma}] \left(\prod_{i' \neq i} \pi^{(i')}[\sigma^{(i')}(z)] \right)$$

$$= \begin{cases} u^{(i)}(z) p_c(z) & \text{if } \pi^{(i)}[\hat{\sigma}] = 1, \pi^{(i')}[\sigma^{(i')}(z)] = 1 \; \forall i' \neq i \\ 0 & \text{otherwise.} \end{cases}$$

Substituting the above expression into (18), we obtain

$$\epsilon \geq \sum_{z \in \mathcal{Z}^{(i)}_j} u^{(i)}(z) p_c(z) \left(\sum_{\substack{(\pi^{(1)}, \ldots, \pi^{(n)}) \in \Pi \\ \pi^{(i')}[\sigma^{(i')}(z)] = 1 \; \forall i' \neq i \\ \pi^{(i)}[\hat{\sigma}] = 1}} \mu[(\pi^{(1)}, \ldots, \pi^{(n)})] \cdot \left(\hat{\pi}[\sigma^{(i)}(z)] - \pi^{(i)}[\sigma^{(i)}(z)] \right) \right)$$

$$= \sum_{z \in \mathcal{Z}^{(i)}_j} u^{(i)}(z) p_c(z) r^{(i)}_{\mu, \hat{\sigma} \to \hat{\pi}}(z) - \sum_{z \in \mathcal{Z}^{(i)}_j} u^{(i)}(z) p_c(z) \left(\sum_{\substack{(\pi^{(1)}, \ldots, \pi^{(n)}) \in \Pi \\ \pi^{(i')}[\sigma^{(i')}(z)] = 1 \; \forall i' \in [n] \\ \pi^{(i)}[\hat{\sigma}] = 1}} \mu[(\pi^{(1)}, \ldots, \pi^{(n)})] \right),$$

where, to get the last equality, we used the definition of $r^{(i)}_{\mu, \hat{\sigma} \to \hat{\pi}}$ and we dropped the factor $\pi^{(i)}[\sigma^{(i)}(z)] \in \{0, 1\}$ in the second summation by adding the condition $\pi^{(i)}[\sigma^{(i)}(z)] = 1$ to its domain. Notice that, by definition of $\mathcal{Z}^{(i)}_j$, the first summation above is exactly the term appearing in the left-hand side of the inequality in the definition of ϵ-EFCE (Definition 2.7). Moreover, since for any $(\pi^{(1)}, \ldots, \pi^{(n)}) \in \Pi$ it holds that $\pi^{(i)}[\sigma^{(i)}(z)] = 1$ and $\pi^{(i)}[\hat{\sigma}] = 1$ only for terminal nodes $z \in \mathcal{Z}^{(i)}_j$ such that $\hat{\sigma} \preceq \sigma^{(i)}(z)$, we can restrict the domain of the second summation above to $z \in \mathcal{Z} : \sigma^{(i)}(z) \succeq \hat{\sigma}$ and equivalently rewrite it as

$$\sum_{\substack{z \in \mathcal{Z} \\ \sigma^{(i)}(z) \succeq \hat{\sigma}}} u^{(i)}(z) p_c(z) \left(\sum_{\substack{(\pi^{(1)}, \ldots, \pi^{(n)}) \in \Pi \\ \pi^{(i')}[\sigma^{(i')}(z)] = 1 \; \forall i' \in [n]}} \mu[(\pi^{(1)}, \ldots, \pi^{(n)})] \right) = \sum_{\substack{z \in \mathcal{Z} \\ \sigma^{(i)}(z) \succeq \hat{\sigma}}} u^{(i)}(z) p_c(z) r_\mu(z),$$

which is exactly the first term appearing in the right-hand side of the inequality in the definition of ϵ-EFCE (Definition 2.7). Thus, we obtain that

$$\epsilon \geq \sum_{\substack{z \in \mathcal{Z} \\ \sigma^{(i)}(z) \succeq j}} u^{(i)}(z) p_c(z) r^{(i)}_{\mu, \hat{\sigma} \to \hat{\pi}}(z) - \sum_{\substack{z \in \mathcal{Z} \\ \sigma^{(i)}(z) \succeq \hat{\sigma}}} u^{(i)}(z) p_c(z) r_\mu(z),$$

for all $\hat{\sigma} = (j, a) \in \Sigma^{(i)}_*$ and $\hat{\pi} \in \Pi^{(i)}_j$, which is the definition of μ being an ϵ-EFCE.   □

4   EFFICIENT NO-TRIGGER-REGRET ALGORITHM

Theorem 3.8 in Section 3 immediately implies that if all players $i \in [n]$ play according to the strategies output by a $\Psi^{(i)}$-regret minimizer for the set of their deterministic sequence-form strategies $\Pi^{(i)}$, then their empirical frequency of play converges to the set of EFCEs. Therefore, the existence of uncoupled no-regret learning dynamics that converge to EFCE can be proved constructively by showing that one such $\Psi^{(i)}$-regret minimizer can be constructed for each player $i \in [n]$. More precisely, in this section, we seek to solve the following problem:

Problem 1. Given any player $i \in [n]$, construct a $\Psi^{(i)}$-regret minimizer for the set of the player's deterministic sequence-form strategies $\Pi^{(i)}$, such that:

- it is efficient: The NextElement and the ObserveUtility operations both run in polynomial time in the number $|\Sigma^{(i)}|$ of sequences of the player; and
- it guarantees low regret: After any T observed linear utility functions and for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ the cumulative $\Psi^{(i)}$-regret is $O(\text{poly}(|\Sigma^{(i)}|) \cdot \sqrt{T \log(1/\delta)})$.

The central result of this section, Theorem 4.17, establishes that the $\Psi^{(i)}$-regret minimizer $\mathcal{R}^{(i)}$ defined in Algorithm 6 satisfies all the requirements of Problem 1.

4.1   Overview

Before delving into the details of the construction of our $\Psi^{(i)}$-regret minimizer for the set of deterministic sequence-form strategies $\Pi^{(i)}$ of a generic player $i \in [n]$, we give an overview of the main logical steps that we use to solve Problem 1.

- In Section 4.2, we show that one can soundly move the attention from the set of *deterministic* sequence-form strategies $\Pi^{(i)}$ to the set of *mixed* sequence-form strategies $Q^{(i)} = \text{co}(\Pi^{(i)})$. In particular, in the rest of the section, we will seek to construct a $\Psi^{(i)}$-regret minimizer for the set $Q^{(i)}$ (as opposed to $\Pi^{(i)}$) that guarantees sublinear regret and polynomial-time implementation in the worst case.
- In Section 4.3, we show that the convex hull $\text{co}(\Psi^{(i)})$ of the set of canonical trigger deviation functions possesses a combinatorial structure that can be leveraged to construct an efficient deterministic regret minimizer for it.
- Finally, in Section 4.4, we prove that given any $\phi \in \text{co}(\Psi^{(i)})$, there exists a fixed-point sequence-form strategy $q \in Q^{(i)}$ such that $\phi(q) = q$, and that such a fixed-point strategy can be found in polynomial time in the number of sequences $|\Sigma^{(i)}|$ of Player i.

Together, the last two steps enable us to apply the construction by Gordon et al. [20] described in Section 2.4 to obtain an efficient $\text{co}(\Psi^{(i)})$-regret minimizer for the set of sequence-form strategies $Q^{(i)}$ with worst-case sublinear regret guarantees. We summarize that construction in Figure 4, which can serve as a reading aid for the section. Since $\text{co}(\Psi^{(i)}) \supseteq \Psi^{(i)}$, such $\text{co}(\Psi^{(i)})$-regret minimizer is also a $\Psi^{(i)}$-regret minimizer, and the construction is complete.

4.2 From Deterministic to Mixed Strategies

Suppose that a regret minimizer for a generic discrete set $\mathcal{X}$ were sought, but only regret minimizers for the convex hull co($\mathcal{X}$) were known. It seems natural to wonder whether one could take any regret minimizer for co($\mathcal{X}$) and convert it into a regret minimizer for $\mathcal{X}$ by sampling the outputs $\bar{x}^t$ of the former using an unbiased estimator $x^t \in \mathcal{X}$, with $\mathbb{E}[x^t] = \bar{x}^t$. It is a folklore result, justified by a concentration argument, that this is indeed the case (see, for instance, Reference [9, page 192]). In particular, in the case of our interest where $\mathcal{X} = \Pi^{(i)}$, the following can be shown:

LEMMA 4.1. *Let* $i \in [n]$ *be any player, and* $\bar{\mathcal{R}}^{(i)}$ *be any* $\Psi^{(i)}$-*regret minimizer for the set* $Q^{(i)}$ *of* mixed *sequence-form strategies. Consider the algorithm* $\mathcal{R}^{(i)}$ *whose* NEXTELEMENT *and* OBSERVEUTILITY *operations are defined as follows at all times* t:

- $\mathcal{R}^{(i)}$.NEXTELEMENT *calls* $\bar{\mathcal{R}}^{(i)}$.NEXTELEMENT, *thereby obtaining a mixed sequence-form strategy* $q^{(i),\,t} \in Q^{(i)}$. *Then, an unbiased sampling scheme (such as the natural sampling scheme described in Section 2.2) is used to sample a random deterministic sequence-form strategy* $\pi^{(i),\,t} \in \Pi^{(i)}$ *in linear time in the number of sequences* $|\Sigma^{(i)}|$. *Finally,* $\pi^{(i),\,t}$ *is returned to the caller;*
- $\mathcal{R}^{(i)}$.OBSERVEUTILITY(ℓ^t) *calls* $\bar{\mathcal{R}}^{(i)}$.OBSERVEUTILITY(ℓ^t) *with the same utility function* ℓ^t.

Furthermore, assume that the linear utility functions $\ell^1, \ell^2, \ldots$ *received as feedback by* $\mathcal{R}^{(i)}$ *have range upper bounded by a constant* D, *that is,* $\max_{q,\,q' \in Q^{(i)}}\{\ell^t(q) - \ell^t(q')\} \le D$ *for all* $t = 1, \ldots, T$. *Then,* $\mathcal{R}^{(i)}$ *is a* $\Psi^{(i)}$-*regret minimizer for the set of deterministic sequence-form strategies* $\Pi^{(i)}$, *and its cumulative* $\Psi^{(i)}$-*regret satisfies, at all times* T *and for all* $\delta \in (0, 1)$, *the inequality*

$$\mathbb{P}\left[R^{(i),\,T} \le \bar{R}^{(i),\,T} + 4D\sqrt{T \cdot |\Sigma^{(i)}| \log\left(\frac{1}{\delta}\right)}\right] \ge 1 - \delta.$$

PROOF. Let $\ell^1, \ell^2, \ldots$ be the sequence of linear utility functions observed by $\mathcal{R}^{(i)}$, and fix any $\phi \in \Psi^{(i)}$. We introduce the discrete-time stochastic process[8]

$$w^t := \ell^t(\phi(\pi^{(i),\,t})) - \ell^t(\pi^{(i),\,t}) - \ell^t(\phi(q^{(i),\,t})) + \ell^t(q^{(i),\,t}). \tag{19}$$

Since (i) ℓ^t and ϕ are both linear functions, (ii) ℓ^t is independent[9] of $\pi^{(i),\,t}$, and (iii) $\pi^{(i),\,t}$ is an unbiased estimator of $q^{(i),\,t}$ at all times t by hypothesis, then w^t is a martingale difference sequence. Furthermore, each increment $|w^t|$ can be easily upper bounded, at all times t, according to

$$|w^t| \le |\ell^t(\phi(\pi^{(i),\,t})) - \ell^t(\pi^{(i),\,t})| + |\ell^t(\phi(q^{(i),\,t})) - \ell^t(q^{(i),\,t})| \le 2D, \tag{20}$$

where the second inequality follows from the fact that ϕ maps sequence-form strategies to sequence-form strategies, together with the hypothesis that ℓ^t has range upper bounded by D.

For any T, let $R^{(i),\,T}(\phi)$ and $\bar{R}^{(i),\,T}(\phi)$ denote the regret cumulated by $\mathcal{R}^{(i)}$ and $\bar{\mathcal{R}}^{(i)}$, respectively, compared to always picking transformation ϕ; in symbols

$$R^{(i),\,T}(\phi) := \sum_{t=1}^{T} \ell^t(\phi(\pi^{(i),\,t})) - \ell^t(\pi^{(i),\,t}), \qquad \bar{R}^{(i),\,T}(\phi) := \sum_{t=1}^{T} \ell^t(\phi(q^{(i),\,t})) - \ell^t(q^{(i),\,t}).$$

---

[8]As is common in the analysis of randomized algorithms, the stochastic process is adapted to the filtration of σ-algebras generated by all the past random outcomes of the algorithm (in our case, all past outcomes of the sources of randomness used by the regret minimizer and by the environment). In other words, using the notation of Footnote 5, $\{w^t\}$ is adapted to the filtration $\{\mathcal{F}^t\}$.

[9]More precisely, ℓ^t is conditionally independent of $\pi^{(i),\,t}$, given $\mathcal{F}^{t-1}$ (cf. Definition 2.8 and Footnote 5).
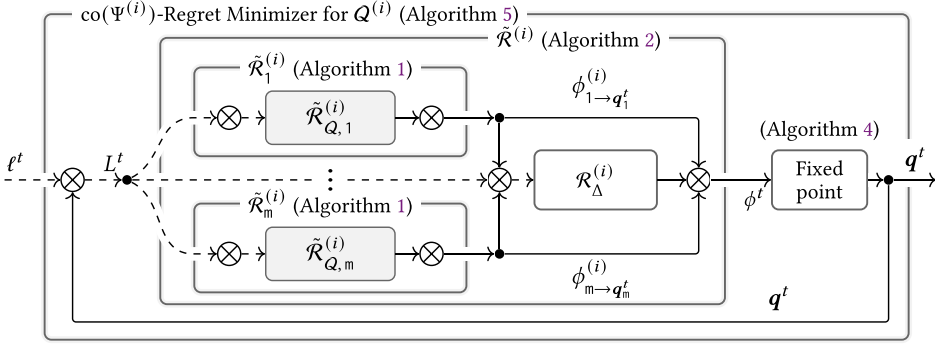
Fig. 4. Pictorial depiction of our $\mathrm{co}(\Psi^{(i)})$-regret minimizer for the set of sequence-form strategies $Q^{(i)}$. The symbol $\otimes$ in the figure denotes a multilinear transformation of the input(s) into the output. Dashed lines denote utility functions. For notational convenience, we let $\Sigma_*^{(i)} := \{1, \ldots, \mathfrak{m}\}$.

It is immediate to see from Definition (19) of w^t, that

$$\sum_{t=1}^{T} w^t = R^{(i),\,T}(\phi) - \bar{R}^{(i),\,T}(\phi) \qquad \forall\, T \in \{1, 2, \ldots\}. \tag{21}$$

Using the Azuma-Hoeffding concentration inequality,[10] it follows that, for all T,

$$\mathbb{P}\left[R^{(i),\,T}(\phi) - \bar{R}^{(i),\,T}(\phi) \leq D\sqrt{8T\log\left(\frac{1}{\delta}\right)}\right] = \mathbb{P}\left[\sum_{t=1}^{T} w^t \leq D\sqrt{8T\log\left(\frac{1}{\delta}\right)}\right]$$

$$\geq 1 - \exp\left\{-\frac{2}{\sum_{t=1}^{T}(2|w^t|)^2}\left(D\sqrt{8T\log\left(\frac{1}{\delta}\right)}\right)^2\right\}$$

$$\geq 1 - \exp\left\{-\frac{2}{(4D)^2 T}\left(D\sqrt{8T\log\left(\frac{1}{\delta}\right)}\right)^2\right\} = 1 - \delta, \quad (22)$$

where we used (21) in the equality and (20) in the second inequality. Since (22) holds for any choice of $\phi \in \Psi^{(i)}$, we can now write

$$\mathbb{P}\left[R^{(i),\,T} \leq \bar{R}^{(i),\,T} + D\sqrt{8T\log\left(\frac{1}{\delta}\right)}\right] = \mathbb{P}\left[\max_{\phi \in \Psi^{(i)}}\{R^{(i),\,T}(\phi)\} \leq \max_{\phi \in \Psi^{(i)}}\{\bar{R}^{(i),\,T}(\phi)\} + D\sqrt{8T\log\left(\frac{1}{\delta}\right)}\right]$$

$$\geq \mathbb{P}\left[\max_{\phi \in \Psi^{(i)}}\{R^{(i),\,T}(\phi) - \bar{R}^{(i),\,T}(\phi)\} \leq D\sqrt{8T\log\left(\frac{1}{\delta}\right)}\right]$$

---

[10]We recall the classic Azuma-Hoeffding inequality [2, 26] for martingale difference sequences (e.g., Reference [34, Theorem 3.14]).

LEMMA 4.2 (AZUMA-HOEFFDING INEQUALITY). *Let $Y_1, \ldots, Y_n$ be a martingale difference sequence with $a_k \leq Y_k \leq b_k$ for each k, for suitable constants a_k, b_k. Then for any $\tau \geq 0$, $\mathbb{P}[\sum Y_k \leq \tau] \geq 1 - e^{-2\tau^2/\sum(b_k - a_k)^2}$.*

$$= \mathbb{P}\left[R^{(i),\,T}(\phi) - \bar{R}^{(i),\,T}(\phi) \le D\sqrt{8T\log\left(\frac{1}{\delta}\right)} \quad \forall\,\phi \in \Psi^{(i)}\right]$$

$$\ge 1 - |\Psi^{(i)}| \cdot \delta, \tag{23}$$

where the first inequality follows from the fact that $\max_{\phi \in \Psi^{(i)}}\{R^{(i),\,T}(\phi)\} - \max_{\phi \in \Psi^{(i)}}\{\bar{R}^{(i),\,T}(\phi)\} \le \max_{\phi \in \Psi^{(i)}}\{R^{(i),\,T}(\phi) - \bar{R}^{(i),\,T}(\phi)\}$, while the second inequality follows from applying (22) and the union bound. Using Lemma 3.5, setting $\delta' := |\Sigma^{(i)}| \cdot 2^{|\Sigma^{(i)}|} \cdot \delta$, and noticing that $\log(|\Sigma^{(i)}| \cdot 2^{|\Sigma^{(i)}|}) \le \log(2^{2|\Sigma^{(i)}|}) \le 2|\Sigma^{(i)}|$, we obtain the statement.   □

Lemma 4.1 immediately implies that, to solve Problem 1, it is enough to solve the following problem:

Problem 2. Given any player $i \in [n]$, construct a $\Psi^{(i)}$-regret minimizer for the set of mixed sequence-form strategies $Q^{(i)}$ such that:

- it is efficient: Both the NEXTELEMENT and the OBSERVEUTILITY operations can be implemented in polynomial time in $|\Sigma^{(i)}|$; and
- it guarantees low regret: After any T observed utilities, the cumulative $\Psi^{(i)}$-regret is upper bounded as $O(\text{poly}(|\Sigma^{(i)}|) \cdot \sqrt{T})$.

The remainder of this section gives an algorithm that solves Problem 2 and, thus, indirectly, also Problem 1.

4.3 Regret Minimizer for the Convex Hull of the Set of Trigger Deviation Functions

In this subsection, we begin the construction of a phi-regret minimizer relative to the convex hull $\text{co}(\Psi^{(i)})$ of the set of trigger deviation functions $\Psi^{(i)}$ for the set $Q^{(i)}$. Since $\text{co}(\Psi^{(i)}) \supseteq \Psi^{(i)}$, any such $(\text{co}(\Psi^{(i)}))$-regret minimizer is trivially also a $\Psi^{(i)}$-regret minimizer for $Q^{(i)}$.

To obtain our $(\text{co}(\Psi^{(i)}))$-regret minimizer, we will leverage the general framework due to Gordon et al. [20] that we recalled at the end of Section 2.4. In our particular case, that construction reduces to showing the following:

(1) existence of a deterministic regret minimizer for the set of deviations $\text{co}(\Psi^{(i)})$; and
(2) existence of a fixed point $q = \phi(q)$ for any $\phi \in \text{co}(\Psi^{(i)})$.

In this subsection, we will focus on point (1), while in the next subsection, we will focus on point (2). Specifically, the central result of this subsection, Theorem 4.6, will constructively establish the existence of an efficient deterministic regret minimizer $\tilde{\mathcal{R}}^{(i)}$ for the set $\text{co}(\Psi^{(i)})$.

The starting point of our approach is the observation that, because the convex hull operation is associative, $\text{co}(\Psi^{(i)}) = \text{co}(\{\phi^{(i)}_{\hat{\sigma} \to \hat{\pi}} : \hat{\sigma} = (j, a) \in \Sigma^{(i)}_*, \hat{\pi} \in \Pi^{(i)}_j\})$ can be evaluated in two stages: First, for each sequence $\hat{\sigma} = (j, a) \in \Sigma^{(i)}_*$ one can define the set

$$\bar{\Psi}^{(i)}_{\hat{\sigma}} := \text{co}\left(\left\{\phi^{(i)}_{\hat{\sigma} \to \hat{\pi}} : \hat{\pi} \in \Pi^{(i)}_j\right\}\right);$$

and then, one can take the convex hull of all $\bar{\Psi}^{(i)}_{\hat{\sigma}}$, that is,

$$\text{co}(\Psi^{(i)}) = \text{co}\left(\left\{\bar{\Psi}^{(i)}_{\hat{\sigma}} : \hat{\sigma} \in \Sigma^{(i)}_*\right\}\right). \tag{24}$$

Our construction of $\tilde{\mathcal{R}}^{(i)}$ will follow a similar structure. First, for each $\hat{\sigma} \in \Sigma^{(i)}_*$, we will construct a regret minimizer $\tilde{\mathcal{R}}^{(i)}_{\hat{\sigma}}$ for the set of deviations $\bar{\Psi}^{(i)}_{\hat{\sigma}}$ (Section 4.3.1). Then, we will combine all the regret minimizers $\tilde{\mathcal{R}}^{(i)}_{\hat{\sigma}}$ into a composite regret minimizer $\tilde{\mathcal{R}}^{(i)}$ for the set $\text{co}(\Psi^{(i)})$ (Section 4.3.2).

4.3.1 Regret Minimizer for $\bar{\Psi}_{\hat{\sigma}}^{(i)}$. Fix any $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$. A deterministic regret minimizer for the set $\bar{\Psi}_{\hat{\sigma}}^{(i)}$ can be constructed starting from any deterministic regret minimizer for the set $Q_j^{(i)}$. The crucial insight lies in the observation that the mapping

$$h_{\hat{\sigma}}^{(i)} : \mathbb{R}^{\Sigma_j^{(i)}} \ni \boldsymbol{y} \mapsto \phi_{\hat{\sigma} \to \boldsymbol{y}}^{(i)}$$

is affine, since the entries in $\boldsymbol{M}_{\hat{\sigma} \to \boldsymbol{y}}^{(i)}$ are defined using only constants and linear combinations of entries in $\boldsymbol{y}$ (Definition 3.2). Hence, using the properties of affine functions, we can write

$$\mathrm{co}\left(\left\{ \phi_{\hat{\sigma} \to \hat{\boldsymbol{\pi}}}^{(i)} : \hat{\boldsymbol{\pi}} \in \Pi_j^{(i)} \right\} \right) = \mathrm{co}(h_{\hat{\sigma}}^{(i)}(\Pi_j^{(i)})) = h_{\hat{\sigma}}^{(i)}(\mathrm{co}(\Pi_j^{(i)})) = h_{\hat{\sigma}}^{(i)}(Q_j^{(i)}).$$

So, we have just proved the following characterization of the set $\bar{\Psi}_{\hat{\sigma}}^{(i)}$:

LEMMA 4.3. *For all sequences $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$, $\bar{\Psi}_{\hat{\sigma}}^{(i)}$ is the image of $Q_j^{(i)}$ under the affine mapping $h_{\hat{\sigma}}^{(i)}$. In symbols,*

$$\bar{\Psi}_{\hat{\sigma}}^{(i)} = \left\{ \phi_{\hat{\sigma} \to \boldsymbol{q}_{\hat{\sigma}}}^{(i)} : \boldsymbol{q}_{\hat{\sigma}} \in Q_j^{(i)} \right\}.$$

As a consequence of Lemma 4.3, given any $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$, all transformations $\phi \in \bar{\Psi}_{\hat{\sigma}}^{(i)}$ are of the form $\phi = \phi_{\hat{\sigma} \to \boldsymbol{q}_{\hat{\sigma}}}^{(i)}$ for some $\boldsymbol{q}_{\hat{\sigma}} \in Q_j^{(i)}$. Thus, the cumulative regret incurred by a generic sequence of transformations $\phi^1 = \phi_{\hat{\sigma} \to \boldsymbol{q}_{\hat{\sigma}}^1}^{(i)}, \dots, \phi^T = \phi_{\hat{\sigma} \to \boldsymbol{q}_{\hat{\sigma}}^T}^{(i)}$ against generic linear utility functions $L^1, \dots, L^T$ can be written as

$$\max_{\phi^* \in \bar{\Psi}_{\hat{\sigma}}^{(i)}} \sum_{t=1}^{T} L^t(\phi^*) - L^t(\phi^t) = \max_{\hat{\boldsymbol{q}}^* \in Q_j^{(i)}} \sum_{t=1}^{T} L^t(\phi_{\hat{\sigma} \to \hat{\boldsymbol{q}}^*}^{(i)}) - L^t(\phi_{\hat{\sigma} \to \boldsymbol{q}_{\hat{\sigma}}^t}^{(i)})$$

$$= \max_{\hat{\boldsymbol{q}}^* \in Q_j^{(i)}} \sum_{t=1}^{T} (L^t \circ h_{\hat{\sigma}}^{(i)})(\hat{\boldsymbol{q}}^*) - (L^t \circ h_{\hat{\sigma}}^{(i)})(\boldsymbol{q}_{\hat{\sigma}}^t). \tag{25}$$

Since L^t is linear and $h_{\hat{\sigma}}^{(i)}$ is affine, their composition $L^t \circ h_{\hat{\sigma}}^{(i)}$ is affine, and therefore the shifted function

$$g_{\hat{\sigma}}^{(i), t} : \mathbb{R}^{\Sigma_j^{(i)}} \ni \boldsymbol{x} \mapsto L^t(h_{\hat{\sigma}}^{(i)}(\boldsymbol{x})) - L^t(h_{\hat{\sigma}}^{(i)}(\boldsymbol{0}))$$

is linear.[11] Furthermore, from (25) it follows that

$$\max_{\phi^* \in \bar{\Psi}_{\hat{\sigma}}^{(i)}} \sum_{t=1}^{T} L^t(\phi^*) - L^t(\phi_{\hat{\sigma} \to \boldsymbol{q}_{\hat{\sigma}}^t}^{(i)}) = \max_{\hat{\boldsymbol{q}}^* \in Q_j^{(i)}} \sum_{t=1}^{T} g_{\hat{\sigma}}^{(i), t}(\hat{\boldsymbol{q}}^*) - g_{\hat{\sigma}}^{(i), t}(\boldsymbol{q}_{\hat{\sigma}}^t). \tag{26}$$

Equation (26) suggests that if the continuation strategies $\boldsymbol{q}_{\hat{\sigma}}^t \in Q_j^{(i)}$ are picked by a deterministic regret minimizer $\tilde{\mathcal{R}}_{Q, \hat{\sigma}}^{(i)}$ that observes the linear utility functions $g_{\hat{\sigma}}^{(i), t}$ at all times t, then the regret cumulated with respect to utility functions L^t by the corresponding elements $\phi_{\hat{\sigma} \to \boldsymbol{q}_{\hat{\sigma}}^t}^{(i)}$ grows sublinearly. We make that construction explicit in Algorithm 1.

---

[11]We shift $L^t \circ h_{\hat{\sigma}}^{(i)}$ purely for technical reasons. We do it so $g^{(i), t}$ is a *linear* utility function, and thus it can be passed in as feedback to a regret minimizer.

ALGORITHM 1: Regret minimizer $\tilde{\mathcal{R}}_{\hat{\sigma}}^{(i)}$ for the set $\bar{\Psi}_{\hat{\sigma}}^{(i)} \coloneqq \{\phi_{\hat{\sigma}\to q_{\hat{\sigma}}}^{(i)} : q_{\hat{\sigma}} \in Q_j^{(i)}\}$

    Data: • $i \in [n]$ player
                  • $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$ trigger sequence
                  • $\tilde{\mathcal{R}}_{Q,\hat{\sigma}}^{(i)}$ deterministic regret minimizer for set $Q_j^{(i)}$ (e.g., the CFR algorithm [49])

1 **function** NextElement()
2     $q_{\hat{\sigma}}^t \leftarrow \tilde{\mathcal{R}}_{Q,\hat{\sigma}}^{(i)}$.NextElement()
3   **return** $\phi_{\hat{\sigma}\to q_{\hat{\sigma}}^t}^{(i)}$, *represented in memory implicitly through the vector* $q_{\hat{\sigma}}^t$

4 **function** ObserveUtility(L^t)
5     $g_{\hat{\sigma}}^{(i),t} \leftarrow$ linear function $\mathbb{R}^{\Sigma_j^{(i)}} \ni x \mapsto L^t(h_{\hat{\sigma}}^{(i)}(x)) - L^t(h_{\hat{\sigma}}^{(i)}(\mathbf{0}))$
6     $\tilde{\mathcal{R}}_{Q,\hat{\sigma}}^{(i)}$.ObserveUtility($g_{\hat{\sigma}}^{(i),t}$)

Algorithm 1 can be instantiated with any deterministic regret minimizer $\tilde{\mathcal{R}}_{Q,\hat{\sigma}}^{(i)}$ for the set of sequence-form strategies $Q_j^{(i)}$. The following proposition formalizes the cumulative regret guarantee when $\tilde{\mathcal{R}}_{Q,\hat{\sigma}}^{(i)}$ is set to the CFR algorithm [49], which so far has arguably been the most widely used regret minimizer for sequence-form strategy spaces:

PROPOSITION 4.4. *Let $i \in [n]$ be any player and $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$ be any trigger sequence. Consider the deterministic regret minimizer $\tilde{\mathcal{R}}_{\hat{\sigma}}^{(i)}$ (Algorithm 1), where $\tilde{\mathcal{R}}_{Q,\hat{\sigma}}$ is set to be the CFR regret minimizer [49]. Upon observing a sequence of linear utility functions $L^1, \ldots, L^T : co(\Psi^{(i)}) \to \mathbb{R}$, the regret cumulated by the elements $\phi^1 = \phi_{\hat{\sigma}\to q_{\hat{\sigma}}^1}^{(i)}, \ldots, \phi^T = \phi_{\hat{\sigma}\to q_{\hat{\sigma}}^T}^{(i)}$ output by $\tilde{\mathcal{R}}_{\hat{\sigma}}^{(i)}$ satisfies*

$$R^T = \max_{\phi^* \in \bar{\Psi}_{\hat{\sigma}}^{(i)}} \sum_{t=1}^T L^t(\phi^*) - L^t(\phi^t) \le D \, |\Sigma_j^{(i)}| \, \sqrt{T},$$

where D is the range of $L^1, \ldots, L^t$, that is, any constant such that $\max_{\phi, \phi' \in \bar{\Psi}_{\hat{\sigma}}^{(i)}} \{L^t(\phi) - L^t(\phi')\} \le D$ for all $t = 1, \ldots, T$. Furthermore, the NextElement and the ObserveUtility operations run in $O(|\Sigma^{(i)}|)$ time.

PROOF. As shown in (26), the regret cumulated by $\tilde{\mathcal{R}}_{\hat{\sigma}}^{(i)}$ upon observing linear utility functions $L^1, \ldots, L^t$ equals the regret cumulated by the CFR algorithm upon observing linear utility functions $g_{\hat{\sigma}}^{(i),t} : \mathbb{R}^{\Sigma_j^{(i)}} \ni x \mapsto L^t(h_{\hat{\sigma}}^{(i)}(x)) - L^t(h_{\hat{\sigma}}^{(i)}(\mathbf{0}))$. Furthermore, the range of $g_{\hat{\sigma}}^{(i),t}$ satisfies the inequality

$$\max_{q, q' \in Q_j^{(i)}} \left\{ g_{\hat{\sigma}}^{(i),t}(q) - g_{\hat{\sigma}}^{(i),t}(q') \right\} = \max_{q, q' \in Q_j^{(i)}} \left\{ L^t(h_{\hat{\sigma}}^{(i)}(q)) - L^t(h_{\hat{\sigma}}^{(i)}(q')) \right\}$$

$$= \max_{\phi, \phi' \in \bar{\Psi}_{\hat{\sigma}}^{(i)}} \left\{ L^t(\phi) - L^t(\phi') \right\} \le D.$$

So, applying the regret bound of the CFR algorithm (Theorems 3 and 4 of Zinkevich et al. [49]),

$$R^T \le D \left(\sum_{j' \succeq j} \sqrt{|\mathcal{A}(j')|} \right) \sqrt{T} \le D \left(\sum_{j' \succeq j} |\mathcal{A}(j')| \right) \sqrt{T} = D |\Sigma_j^{(i)}| \sqrt{T},$$

completing the proof of the regret bound.

The complexity analysis of the NextElement operation follows directly from the fact that CFR's NextElement operation runs in linear time in $|\Sigma_j^{(i)}|$. So, we focus on the complexity of the ObserveUtility operation. Fix any time t, and let $\Lambda^t := \langle L^t \rangle$ be the canonical representation (defined in Section 2.1) of the linear utility function L^t. Since the canonical representation of $h_{\hat{\sigma}}^{(i)}(\boldsymbol{x})$ is the matrix $\boldsymbol{M}_{\hat{\sigma} \to \boldsymbol{x}}^{(i)}$ for all $\boldsymbol{x} \in \mathbb{R}_{\geq 0}^{\Sigma_j^{(i)}}$, from (1), we obtain

$$L^t(h_{\hat{\sigma}}^{(i)}(\boldsymbol{x})) - L^t(h_{\hat{\sigma}}^{(i)}(\boldsymbol{0})) = \sum_{\sigma_r, \sigma_c \in \Sigma_j^{(i)}} \Lambda^t[\sigma_r, \sigma_c] \left(\boldsymbol{M}_{\hat{\sigma} \to \boldsymbol{x}}^{(i)}[\sigma_r, \sigma_c] - \boldsymbol{M}_{\hat{\sigma} \to \boldsymbol{0}}^{(i)}[\sigma_r, \sigma_c] \right)$$

$$= \sum_{\sigma_r \geq j} \Lambda^t[\sigma_r, \hat{\sigma}] \, \boldsymbol{x}[\sigma_r],$$

where the second equality follows from expanding the definitions of $\boldsymbol{M}_{\hat{\sigma} \to \boldsymbol{x}}^{(i)}[\sigma_r, \sigma_c]$ and $\boldsymbol{M}_{\hat{\sigma} \to \boldsymbol{0}}^{(i)}[\sigma_r, \sigma_c]$ given in (8). So, the canonical representation $\langle g_{\hat{\sigma}}^{(i), t} \rangle$ of $g_{\hat{\sigma}}^{(i), t}$ is the vector $(\Lambda^t[\sigma_r, \hat{\sigma}])_{\sigma_r \in \Sigma_j^{(i)}}$, which can be clearly computed and stored in memory in $O(|\Sigma_j^{(i)}|)$ time. Using the fact that CFR's ObserveUtility operation runs in linear time in $|\Sigma_j^{(i)}|$, the complexity bound of the statement follows. □

4.3.2 Regret Minimizer for $co(\Psi^{(i)})$. Recently, Farina et al. [14] showed that a regret minimizer for a composite set of the form $co(\{X_1, \ldots, X_m\})$ can be constructed by combining any individual regret minimizers for $X_1, \ldots, X_m$ through a construction—called a *regret circuit*—which we describe next.

PROPOSITION 4.5 (FARINA ET AL. [14], SECTION 4.3[12]). *Let* $X_1, \ldots, X_m$ *be a finite collection of sets, and let* $\mathcal{R}_1, \ldots, \mathcal{R}_m$ *be any regret minimizers for them. Furthermore, let* $\mathcal{R}_\Delta$ *be any regret minimizer for the m-simplex* $\Delta^m := \{(\lambda_1, \ldots, \lambda_m) \in \mathbb{R}_{\geq 0}^m, \sum_k \lambda_k = 1\}$. *A regret minimizer* $\mathcal{R}_{co}$ *for the set* $co(\{X_1, \ldots, X_m\})$ *can be constructed starting from* $\mathcal{R}_1, \ldots, \mathcal{R}_m$ *and* $\mathcal{R}_\Delta$ *as follows:*

- $\mathcal{R}_{co}.NextElement$ *calls NextElement on each of the regret minimizers* $\mathcal{R}_1, \ldots, \mathcal{R}_m$, *obtaining elements* $\boldsymbol{x}_1^t, \ldots, \boldsymbol{x}_m^t$. *Then, it calls the NextElement operation on* $\mathcal{R}_\Delta$, *obtaining an element of the simplex* $\boldsymbol{\lambda}^t = (\lambda_1^t, \ldots, \lambda_m^t)$. *Finally, it returns the element*

$$\lambda_1^t \boldsymbol{x}_1^t + \cdots + \lambda_m^t \boldsymbol{x}_m^t \in co(\{X_1, \ldots, X_m\}).$$

- $\mathcal{R}_{co}.ObserveUtility(L^t)$ *forwards the linear utility function* L^t *to each of the regret minimizers* $\mathcal{R}_1, \ldots, \mathcal{R}_m$. *Then, it calls the ObserveUtility operation on* $\mathcal{R}_{co}$ *with the linear utility function* $(\lambda_1, \ldots, \lambda_m) \mapsto L^t(\boldsymbol{x}_1^t)\lambda_1 + \cdots + L^t(\boldsymbol{x}_m^t)\lambda_m$.

In doing so, the regret R_{co}^T *cumulated by* $\mathcal{R}_{co}$ *upon observing any T linear utility functions relates to the regrets* $R_1^T, \ldots, R_m^T, R_\Delta^T$ *cumulated by* $\mathcal{R}_1, \ldots, \mathcal{R}_m, \mathcal{R}_\Delta$, *respectively, according to the inequality*

$$R_{co}^T \leq R_\Delta^T + \max\{R_1^T, \ldots, R_m^T\}. \tag{27}$$

We apply the construction described in Proposition 4.5 to obtain our deterministic regret minimizer $\mathcal{R}^{(i)}$ for the set $co(\Psi^{(i)}) = co(\{\bar{\Psi}_{\hat{\sigma}}^{(i)} : \hat{\sigma} \in \Sigma^{(i)}\})$ starting from the deterministic regret minimizers $\tilde{\mathcal{R}}_{\hat{\sigma}}^{(i)}$ (Algorithm 1), one for each sequence $\hat{\sigma} \in \Sigma_*^{(i)}$, as well as any deterministic regret minimizer $\mathcal{R}_\Delta^{(i)}$ for the simplex $\Delta^{\Sigma_*^{(i)}}$. Pseudocode is given in Algorithm 2.

---

[12]Technically, Farina et al. [14] only prove the bound (27) for the case $m = 2$. However, as mentioned by the authors, the extension to generic $m \in \mathbb{N}_{>0}$ is direct.

ALGORITHM 2: Regret minimizer $\tilde{\mathcal{R}}^{(i)}$ for the set $\text{co}(\Psi^{(i)}) = \text{co}(\{\bar{\Psi}_{\hat{\sigma}}^{(i)} : \hat{\sigma} \in \Sigma^{(i)}\})$

Data: • $i \in [n]$ player
  • $\tilde{\mathcal{R}}_{\hat{\sigma}}^{(i)}$ (one for each $\hat{\sigma} \in \Sigma_*^{(i)}$) regret minimizer for $\bar{\Psi}_{\hat{\sigma}}^{(i)}$ as defined in Algorithm 1
  • $\mathcal{R}_{\Delta}^{(i)}$ deterministic regret minimizer for $\Delta^{\Sigma_*^{(i)}}$ (e.g., regret matching [22])

1 **function** NextElement()
2       $\lambda^t \leftarrow \tilde{\mathcal{R}}_{\Delta}^{(i)}.\text{NextElement}()$
3   **for** $\hat{\sigma} \in \Sigma_*^{(i)}$ **do**
4           $\phi_{\hat{\sigma} \to q_{\hat{\sigma}}^t}^{(i)} \leftarrow \tilde{\mathcal{R}}_{\hat{\sigma}}^{(i)}.\text{NextElement}()$ [▷ Algorithm 1]
5   **return** $\sum_{\hat{\sigma} \in \Sigma_*^{(i)}} \lambda^t[\hat{\sigma}] \phi_{\hat{\sigma} \to q_{\hat{\sigma}}^t}^{(i)}$, *represented in memory implicitly as list* $\{(\lambda^t[\hat{\sigma}], q_{\hat{\sigma}}^t)\}_{\hat{\sigma} \in \Sigma_*^{(i)}}$

---

6 **function** ObserveUtility(L^t)
7   **for** $\hat{\sigma} \in \Sigma_*^{(i)}$ **do**
8           $\tilde{\mathcal{R}}_{\hat{\sigma}}^{(i)}.\text{ObserveUtility}(L^t)$ [▷ Algorithm 1]
9       $\ell_{\lambda}^t \leftarrow$ linear utility function $\lambda \mapsto \sum_{\hat{\sigma} \in \Sigma_*^{(i)}} \lambda[\hat{\sigma}] L^t(\phi_{\hat{\sigma} \to q_{\hat{\sigma}}^t}^{(i)})$
10      $\tilde{\mathcal{R}}_{\Delta}^{(i)}.\text{ObserveUtility}(\ell_{\lambda}^t)$

Combining Proposition 4.5 and Proposition 4.4, we obtain the following result:

THEOREM 4.6. *Consider the regret minimizer $\tilde{\mathcal{R}}^{(i)}$ (Algorithm 2), where $\mathcal{R}_{\Delta}^{(i)}$ is set to the regret matching algorithm, and $\tilde{\mathcal{R}}_{\hat{\sigma}}^{(i)}$ is instantiated as described in Proposition 4.4. Upon observing a sequence of linear utility functions $L^1, \ldots, L^T : \text{co}(\Psi^{(i)}) \to \mathbb{R}$, the regret cumulated by the transformations $\phi^1, \ldots, \phi^T \in \text{co}(\Psi^{(i)})$ output by $\tilde{\mathcal{R}}^{(i)}$ satisfies*

$$R^T = \max_{\phi^* \in \text{co}(\Psi^{(i)})} \sum_{t=1}^{T} L^t(\phi^*) - L^t(\phi^t) \leq 2D |\Sigma^{(i)}| \sqrt{T},$$

where D is the range of $L^1, \ldots, L^t$, that is, any constant such that $\max_{\phi, \phi' \in \text{co}(\Psi^{(i)})} \{L^t(\phi) - L^t(\phi')\} \leq D$ for all $t = 1, \ldots, T$. Furthermore, the NextElement and the ObserveUtility operations run in $O(|\Sigma^{(i)}|^2)$ time.

PROOF. At all t, the range of the linear utility function $\lambda \mapsto \sum_{\hat{\sigma} \in \Sigma_*^{(i)}} \lambda[\hat{\sigma}] L^t(\phi_{\hat{\sigma} \to q_{\hat{\sigma}}^t}^{(i)})$ is upper bounded by D. Hence, from the known regret bound of the regret matching algorithm [22, 49], the regret cumulated by $\mathcal{R}_{\Delta}^{(i)}$ after T iterations is upper bounded as

$$R_{\Delta}^T \leq D\sqrt{|\Sigma_*^{(i)}|}\sqrt{T} \leq D|\Sigma^{(i)}|\sqrt{T}.$$

However, the regret bound in Proposition 4.4 shows that, for all $\hat{\sigma} = (j, a) \in \Sigma_*^{(i)}$, the regret $R_{\hat{\sigma}}^T$ cumulated by $\tilde{\mathcal{R}}_{\hat{\sigma}}^{(i)}$ is upper bounded as $R_{\hat{\sigma}}^T \leq D|\Sigma_j^{(i)}|\sqrt{T}$. Applying (27) together with the fact that $|\Sigma_j^{(i)}| \leq |\Sigma^{(i)}|$ for all $j \in \mathcal{J}^{(i)}$ yields the regret bound in the statement.

The complexity analysis of NextElement is straightforward: The regret matching algorithm produces elements in $O(|\Sigma^{(i)}|)$ time, while each iteration of the loop over $\Sigma_*^{(i)}$ requires $O(|\Sigma^{(i)}|)$ time. Then, we focus on the complexity of ObserveUtility. There, the only operation whose analysis is not immediately obvious is the construction of the linear utility function $\ell_{\lambda}^t : \lambda \mapsto \sum_{\hat{\sigma} \in \Sigma_*^{(i)}} \lambda[\hat{\sigma}] L^t(\phi_{\hat{\sigma} \to q_{\hat{\sigma}}^t}^{(i)})$, where it is necessary to check that its canonical representation

(Section 2.1), given by the vector $(L^t(\phi^{(i)}_{\hat{\sigma} \to \boldsymbol{q}^t_{\hat{\sigma}}}))_{\hat{\sigma} \in \Sigma^{(i)}_*}$, can be computed and stored in memory in $O(|\Sigma^{(i)}|^2)$ time. Fix any $\hat{\sigma} \in \Sigma^{(i)}_*$. The canonical representation of $\phi^{(i)}_{\hat{\sigma} \to \boldsymbol{q}^t_{\hat{\sigma}}}$ is $\boldsymbol{M}^{(i)}_{\hat{\sigma} \to \boldsymbol{q}^t_{\hat{\sigma}}}$, which is a matrix with $O(|\Sigma^{(i)}|)$ nonzero entries. Therefore, using (1), the evaluation of $L^t(\phi^{(i)}_{\hat{\sigma} \to \boldsymbol{q}^t_{\hat{\sigma}}})$ via the canonical representations of L^t (given as input) and $\phi^{(i)}_{\hat{\sigma} \to \boldsymbol{q}^t_{\hat{\sigma}}}$ takes $O(|\Sigma^{(i)}|)$ time. Thus, the representation of ℓ_λ can be computed in $O(|\Sigma^{(i)}|^2)$ time, confirming the analysis in the statement. □

4.4 Computation of the Next Strategy

In this subsection, we complete the construction of our $\mathrm{co}(\Psi^{(i)})$-regret minimizer for $Q^{(i)}$ started in Section 4.3 by showing that every transformation $\phi \in \mathrm{co}(\Psi^{(i)})$ admits a fixed point $\phi(\boldsymbol{q}) = \boldsymbol{q} \in Q^{(i)}$ and that such a fixed point can be computed in time quadratic in the number of sequences $\Sigma^{(i)}$ of Player i.

As a key step in our algorithm, we will use the following well-known result about stationary distributions of stochastic matrices:

LEMMA 4.7. *Let $\boldsymbol{A} \in \mathbb{S}^m$ be a stochastic matrix. Then, $\boldsymbol{A}$ admits a fixed point $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x} \in \Delta^m$. Furthermore, such a fixed point can be computed in polynomial time in m.*

Several algorithms are known for computing fixed points of stochastic matrices (see, e.g., Reference [39] for a comparison of eight different methods). Since the particular choice of method is irrelevant, in this article, we will make the following assumption:

ASSUMPTION 1. *Given any $m \in \mathbb{N}_{>0}$, we assume access to an oracle for computing a fixed point of any $m \times m$ stochastic matrix $\boldsymbol{A}$. Furthermore, we assume that the oracle requires at most $\mathrm{FP}(m)$ time in the worst case to compute any such fixed point.*

Our algorithm for computing a fixed point of $\phi \in \mathrm{co}(\Psi^{(i)})$ requires that the transformation ϕ be expressed as a convex combination of elements from the sets $\{\bar{\Psi}^{(i)}_{\hat{\sigma}}\}_{\hat{\sigma} \in \Sigma^{(i)}_*}$, that is, an expression of the form

$$\phi = \sum_{\hat{\sigma} \in \Sigma^{(i)}_*} \lambda_{\hat{\sigma}} \, \phi^{(i)}_{\hat{\sigma} \to \boldsymbol{q}_{\hat{\sigma}}}, \quad \text{where} \quad \sum_{\hat{\sigma} \in \Sigma^{(i)}_*} \lambda_{\hat{\sigma}} = 1, \text{ and } \lambda_{\hat{\sigma}} \geq 0, \ \boldsymbol{q}_{\hat{\sigma}} \in Q^{(i)}_j \ \forall \, \hat{\sigma} = (j, a) \in \Sigma^{(i)}_*, \quad (28)$$

in accordance with the characterization of $\mathrm{co}(\Psi^{(i)})$ established by (24) and Lemma 4.3. Note that our regret minimizer $\tilde{\mathcal{R}}^{(i)}$ for the set $\mathrm{co}(\Psi^{(i)})$ (Algorithm 2) already outputs transformations ϕ expressed in the form above.

Our algorithm operates *incrementally*, constructing a fixed point sequence-form strategy $\boldsymbol{q}$ for ϕ information set by information set in a top-down fashion. To formalize this notion of top-down construction, we will make use of the two following definitions:

Definition 4.8. Let $i \in [n]$ be a player and $J \subseteq \mathcal{J}^{(i)}$ be a subset of that player's information sets. We say that J is *a trunk of $\mathcal{J}^{(i)}$* if, for every $j \in J$, all predecessors of j (that is, all $j' \in \mathcal{J}^{(i)}$ such that $j' \prec j$) are also in J.

Example 4.9. In the small game of Figure 1 (left), the sets $\{A\}, \{A, B\}, \{A, C\}, \{A, D\}, \{A, B, D\}, \{A, C, D\}$, and $\{A, B, C\ D\} = \mathcal{J}^{(1)}$, as well as the empty seq $\emptyset$, exhaust all the possible trunks for Player 1. Conversely, set $J = \{B\}$ is *not* a trunk for Player 1, because $A \prec B$ and yet $A \notin J$.

Definition 4.10. Let $i \in [n]$ be a player, $J \subseteq \mathcal{J}^{(i)}$ be a trunk of $\mathcal{J}^{(i)}$ (Definition 4.8), and $\phi \in \mathrm{co}(\Psi^{(i)})$. We say that a vector $\boldsymbol{x} \in \mathbb{R}^{\Sigma^{(i)}}_{\geq 0}$ is a *J-partial fixed point of ϕ* if it satisfies the sequence-form

constraints at all $j \in J$, that is,

$$x[\varnothing] = 1, \qquad x[\sigma^{(i)}(j)] = \sum_{a \in \mathcal{A}(j)} x[(j,a)] \quad \forall\, j \in J, \tag{29}$$

and furthermore

$$\phi(x)[\varnothing] = x[\varnothing] = 1, \qquad \phi(x)[(j,a)] = x[(j,a)] \quad \forall\, j \in J,\ a \in \mathcal{A}(j). \tag{30}$$

It follows from Definition 4.10 that a $\mathcal{J}^{(i)}$-partial fixed point of ϕ is a vector $q \in Q^{(i)}$ such that $q = \phi(q)$. The following simple lemma establishes an $\varnothing$-partial fixed point for any transformation $\phi \in \mathrm{co}(\Psi^{(i)})$:

LEMMA 4.11. *Let $i \in [n]$ be a player, and $\phi = \sum_{\hat{\sigma} \in \Sigma_*^{(i)}} \lambda_{\hat{\sigma}} \phi_{\hat{\sigma} \to q_{\hat{\sigma}}}^{(i)}$ be any transformation in $\mathrm{co}(\Psi^{(i)})$, expressed as in* (28). *Then, the vector $x_0 \in \mathbb{R}_{\geq 0}^{\Sigma^{(i)}}$, whose entries are all zeros except for $x_0[\varnothing] = 1$, is a $\varnothing$-partial fixed point of ϕ.*

PROOF. Condition (29) is straightforward. So, we focus on (30). Fix any $\hat{\sigma} = (j,a) \in \Sigma_*^{(i)}$. The definition of $M_{\hat{\sigma} \to \hat{q}}^{(i)}$, given in (8), implies that

$$M_{\hat{\sigma} \to \hat{q}_{\hat{\sigma}}}^{(i)}[\sigma_r, \varnothing] = \begin{cases} 1 & \text{if } \sigma_r = \varnothing \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, \sigma_r \in \Sigma^{(i)}.$$

Consequently, $\phi_{\hat{\sigma} \to \hat{q}_{\hat{\sigma}}}^{(i)}(x_0) = M_{\hat{\sigma} \to \hat{q}_{\hat{\sigma}}}^{(i)} x_0 = x_0$ (from expanding the matrix-vector multiplication). So, $\phi(x_0) = \sum_{\hat{\sigma} \in \Sigma_*^{(i)}} \lambda_{\hat{\sigma}} \phi_{\hat{\sigma} \to \hat{q}_{\hat{\sigma}}}^{(i)}(x_0) = x_0$ and in particular $\phi(x_0)[\varnothing] = x_0[\varnothing] = 1$. So, (30) holds, as we wanted to show.   □

The key result that powers our algorithm to compute a fixed point of any $\phi \in \mathrm{co}(\Psi^{(i)})$ is that a J-partial fixed point can be cheaply promoted to be a $(J \cup \{j^*\})$-partial fixed point, where $j^* \in \mathcal{J}^{(i)} \setminus J$ is any information set whose predecessors are all in J. Algorithm 3 below gives an implementation of such a promotion: EXTEND(ϕ, J, j^*, x) starts with a J-partial fixed point x of ϕ and modifies all entries $x[(j^*, a)]$, $a \in \mathcal{A}(j^*)$, so x becomes a $(J \cup \{j^*\})$-partial fixed point. Therefore, at a conceptual level, one can repeatedly invoke EXTEND, growing the trunk J one information set at a time until $J = \mathcal{J}^{(i)}$, starting from the $\varnothing$-partial fixed point x_0 described in Lemma 4.11.

Before giving a proof of correctness and an analysis of the complexity of EXTEND, we illustrate an application of the algorithm in the simple extensive-form game of Figure 1.

Example 4.12. Consider the simple extensive-form game of Figure 1 (left), and recall the three deviation functions $\phi_{(\text{A},1) \to \hat{\pi}_a}^{(1)}, \phi_{(\text{A},2) \to \hat{\pi}_b}^{(1)}, \phi_{(\text{B},3) \to \hat{\pi}_c}^{(1)}$ considered in Example 3.6. We will illustrate two applications of EXTEND, with respect to the transformation

$$\phi := \frac{1}{2} \phi_{(\text{A},1) \to \hat{\pi}_a}^{(1)} + \frac{1}{3} \phi_{(\text{A},2) \to \hat{\pi}_b}^{(1)} + \frac{1}{6} \phi_{(\text{B},3) \to \hat{\pi}_c}^{(1)} \in \mathrm{co}(\Psi^{(i)})[1].$$

- In the first application, consider the trunk $J = \varnothing$, information set $j^* = \text{A}$, and the $\varnothing$-partial fixed point described in Lemma 4.11, that is, the vector x whose components are all 0 except for the entry corresponding to the empty sequence $\varnothing$, which is set to 1. In this case, $\sigma_p = \sigma^{(i)}(j^*)$ (Line 1 of Algorithm 3) is the empty sequence. Since no information set j' can possibly satisfy $j' \preceq \sigma_p$, the vector r defined on Line 2 is the zero vector. Consequently, the matrix

---

ALGORITHM 3: EXTEND($\phi, J, j^*, \boldsymbol{x}$)

Input : • $\phi = \sum_{\hat{\sigma} \in \Sigma_*^{(i)}} \lambda_{\hat{\sigma}} \phi_{\hat{\sigma} \to \boldsymbol{q}_{\hat{\sigma}}}^{(i)} \in \text{co}(\Psi^{(i)})$ transformation for a player $i \in [n]$, represented in
memory implicitly as the list $\{(\lambda_{\hat{\sigma}}, \boldsymbol{q}_{\hat{\sigma}})\}_{\hat{\sigma} \in \Sigma_*^{(i)}}$

      • $J \subseteq \mathcal{J}^{(i)}$ trunk for Player i

      • $j^* \in \mathcal{J}^{(i)}$ information set not in J such that its immediate predecessor is in J

      • $\boldsymbol{x} \in \mathbb{R}_{\geq 0}^{\Sigma^{(i)}}$ J-partial fixed point of ϕ

Output: • $\boldsymbol{x}' \in \mathbb{R}_{\geq 0}^{\Sigma^{(i)}}$ $(J \cup \{j^*\})$-partial fixed point of ϕ

---

1   $\sigma_p \leftarrow \sigma^{(i)}(j^*)$

2   Let $\boldsymbol{r} \in \mathbb{R}_{\geq 0}^{\mathcal{A}(j^*)}$ be the vector whose entries are defined, for all $a \in \mathcal{A}(j^*)$, as

$$r[a] := \sum_{j' \leq \sigma_p} \sum_{a' \in \mathcal{A}(j')} \lambda_{(j', a')} \boldsymbol{q}_{(j', a')}[(j^*, a)] \boldsymbol{x}[(j', a')]$$

3   Let $\boldsymbol{W} \in \boldsymbol{x}[\sigma_p] \cdot \mathbb{S}^{\mathcal{A}(j^*)}$ be the matrix whose entries are defined, for all $a_r, a_c \in \mathcal{A}(j^*)$, as

$$W[a_r, a_c] := r[a_r] + \left(\lambda_{(j^*, a_c)} \boldsymbol{q}_{(j^*, a_c)}[(j^*, a_r)] \right) + \left(1 - \sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \hat{\sigma} \leq (j^*, a_c)} \lambda_{\hat{\sigma}} \right) \mathbb{1}_{\{a_r = a_c\}} \boldsymbol{x}[\sigma_p]$$

4   **if** $\boldsymbol{x}[\sigma_p] = 0$ **then**

5      $\boldsymbol{w} \leftarrow \boldsymbol{0} \in \mathbb{R}_{\geq 0}^{\mathcal{A}(j^*)}$

6   **else**

7      $\boldsymbol{b} \in \Delta^{\mathcal{A}(j^*)} \leftarrow$ fixed point of stochastic matrix $\frac{1}{\boldsymbol{x}[\sigma_p]} \boldsymbol{W}$

8      $\boldsymbol{w} \leftarrow \boldsymbol{x}[\sigma_p] \boldsymbol{b}$

9   $\boldsymbol{x}' \leftarrow \boldsymbol{x}$

10   **for** $a \in \mathcal{A}(j^*)$ **do**

11      $\boldsymbol{x}'[(j^*, a)] \leftarrow \boldsymbol{w}[(j^*, a)]$

12   **return** $\boldsymbol{x}'$

$\boldsymbol{W}$ defined on Line 3 is

$$\boldsymbol{W} = \begin{array}{cc} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{pmatrix} 1/2 & 1/3 \\ 1/2 & 2/3 \end{pmatrix} \end{array}$$

which is a stochastic matrix. A fixed point for $\boldsymbol{W}$ is given by the vector $\boldsymbol{b} := (2/5, 3/5) \in \Delta^{\{2,3\}}$.
So, the vector $\boldsymbol{x}'$ returned by EXTEND is given by

$$\boldsymbol{x}'[\varnothing] = 1, \quad \boldsymbol{x}'[(\text{A}, 1)] = \frac{2}{5}, \quad \boldsymbol{x}'[(\text{A}, 2)] = \frac{3}{5}$$

and zero entries everywhere else. Direct computation reveals that $\boldsymbol{x}'$ is indeed a $\{\text{A}\}$-partial
fixed point of ϕ.

• In the second application of EXTEND, we start from the $\{\text{A}\}$-partial fixed point that we computed in the previous bullet point and extend it to a $\{\text{A}, \text{D}\}$-partial fixed point. Here, $j^* = \text{D}$,
and so $\sigma_p = (\text{A}, 2)$. The only $j' \leq \sigma_p$ is A, and so the vector $\boldsymbol{r}$ defined on Line 2 is

$$r[7] = \frac{1}{5}, \quad r[8] = 0.$$

Consequently, the matrix W defined on Line 3 is

$$
W = \begin{array}{c} \\ 7 \\ 8 \end{array} \begin{pmatrix} 3/5 & 1/5 \\ 0 & 2/5 \end{pmatrix}
\begin{array}{cc} 7 & 8 \end{array}
$$

As expected, $W \in 3/5\, \mathbb{S}^{\{7,8\}} = x[(A, 2)]\, \mathbb{S}^{\{7,8\}}$. A fixed point for $\frac{1}{x[(A,2)]} W = 5/3\, W$ is given by the vector $b := (1, 0)$. So, the vector x' returned by EXTEND is given by

$$
x'[\varnothing] = 1, \quad x'[(A, 1)] = \frac{2}{5}, \quad x'[(A, 2)] = \frac{3}{5}, \quad x'[(D, 7)] = \frac{3}{5}, \quad x'[(D, 8)] = 0
$$

and zero entries everywhere else. Once again, direct computation reveals that x' is indeed a $\{A, D\}$-partial fixed point of ϕ.

To prove correctness of EXTEND in Proposition 4.14, we will find useful the following technical lemma:

LEMMA 4.13. *Let $i \in [n]$ be any player, and $\phi = \sum_{\hat{\sigma} \in \Sigma_*^{(i)}} \lambda_{\hat{\sigma}} \phi_{\hat{\sigma} \to q_{\hat{\sigma}}}^{(i)}$ be any linear transformation in $\mathrm{co}(\Psi^{(i)})$ expressed as in (28). Then, for all $\sigma \in \Sigma^{(i)}$,*

$$
\phi(x)[\sigma] = \left(1 - \sum_{\hat{\sigma} \in \Sigma_*^{(i)},\, \hat{\sigma} \preceq \sigma} \lambda_{\hat{\sigma}} \right) x[\sigma] + \sum_{j' \preceq \sigma} \sum_{a' \in \mathcal{A}(j')} \lambda_{(j', a')}\, q_{(j', a')}[\sigma]\, x[(j', a')].
$$

PROOF. Fix any trigger sequence $\hat{\sigma} = (j', a') \in \Sigma_*^{(i)}$. By expanding the matrix-vector multiplication between $M_{\hat{\sigma} \to q_{\hat{\sigma}}}^{(i)}$ (Definition 3.2) and x, we have that for all $\sigma \in \Sigma^{(i)}$,

$$
\phi_{\hat{\sigma} \to q_{\hat{\sigma}}}^{(i)}(x)[\sigma] = x[\sigma] \mathbb{1}_{\{\sigma \not\succeq \hat{\sigma}\}} + q_{\hat{\sigma}}[\sigma] x[\hat{\sigma}] \mathbb{1}_{\{\sigma \succeq j'\}}. \tag{31}
$$

Therefore, for all $\sigma \in \Sigma^{(i)}$,

$$
\phi(x)[\sigma] = \sum_{\hat{\sigma} \in \Sigma_*^{(i)}} \lambda_{\hat{\sigma}}\, \phi_{\hat{\sigma} \to q_{\hat{\sigma}}}^{(i)}(x)[\sigma] = \sum_{\hat{\sigma} = (j', a') \in \Sigma_*^{(i)}} \lambda_{\hat{\sigma}} \left(x[\sigma] \mathbb{1}_{\{\sigma \not\succeq \hat{\sigma}\}} + q_{\hat{\sigma}}[\sigma] x[\hat{\sigma}] \mathbb{1}_{\{\sigma \succeq j'\}} \right)
$$

$$
= \left(\sum_{\hat{\sigma} \in \Sigma_*^{(i)},\, \sigma \not\succeq \hat{\sigma}} \lambda_{\hat{\sigma}} \right) x[\sigma] + \sum_{j' \preceq \sigma} \sum_{a' \in \mathcal{A}(j')} \lambda_{(j', a')}\, q_{(j', a')}[\sigma]\, x[(j', a')]
$$

$$
= \left(1 - \sum_{\hat{\sigma} \in \Sigma_*^{(i)},\, \hat{\sigma} \preceq \sigma} \lambda_{\hat{\sigma}} \right) x[\sigma] + \sum_{j' \preceq \sigma} \sum_{a' \in \mathcal{A}(j')} \lambda_{(j', a')}\, q_{(j', a')}[\sigma]\, x[(j', a')],
$$

as we wanted to show. □

PROPOSITION 4.14. *Let $i \in [n]$ be a player, $\phi = \sum_{\hat{\sigma} \in \Sigma_*^{(i)}} \lambda_{\hat{\sigma}} \phi_{\hat{\sigma} \to q_{\hat{\sigma}}}^{(i)}$ be a linear transformation in $\mathrm{co}(\Psi^{(i)})$ expressed as in (28), $x \in \mathbb{R}_{\geq 0}^{\Sigma^{(i)}}$ be a J-partial fixed point of ϕ, and $j^* \in \mathcal{J}^{(i)}$ be a information set not in J such that its immediate predecessor is in J. Then, EXTEND(ϕ, J, j^*, x), given in Algorithm 3, computes a $(J \cup \{j^*\})$-partial fixed point of ϕ in time upper bounded by $O(|\Sigma^{(i)}| |\mathcal{A}(j^*)| + \mathrm{FP}(|\mathcal{A}(j^*)|))$.*

PROOF. We break the proof into four parts. In the first part, we analyze the sum of the entries of vector r defined in Line 2 of Algorithm 3. In the second part, we prove that $\frac{1}{x[\sigma_p]} W \in \mathbb{S}^{\mathcal{A}(j^*)}$, as

stated in Line 3. In the third part, we show that the output $\boldsymbol{x}'$ of the algorithm is indeed a $(J \cup \{j^*\})$-partial fixed point of ϕ. Finally, in the fourth part, we analyze the computational complexity of the algorithm.

Part 1: sum of the entries of $\boldsymbol{r}$. In this first part of the proof, we study the sum of the entries of the vector $\boldsymbol{r}$ defined on Line 2 in Algorithm 3. By hypothesis, the immediate predecessor of j^* is in J. So, because $\boldsymbol{x}$ is a J-partial fixed point, the sequence $\sigma_p := \sigma^{(i)}(j^*)$ satisfies $\phi(\boldsymbol{x})[\sigma_p] = \boldsymbol{x}[\sigma_p]$. Hence, expanding the term $\phi(\boldsymbol{x})[\sigma_p]$ using Lemma 4.13, we conclude that

$$\left(1 - \sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \leq \sigma_p} \lambda_{\hat{\sigma}} \right) \boldsymbol{x}[\sigma_p] + \sum_{j' \leq \sigma_p} \sum_{a' \in \mathcal{A}(j')} \lambda_{(j',a')} \, \boldsymbol{q}_{(j',a')}[\sigma_p] \, \boldsymbol{x}[(j',a')] = \boldsymbol{x}[\sigma_p].$$

By rearranging terms, we have

$$\left(\sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \leq \sigma_p} \lambda_{\hat{\sigma}} \right) \boldsymbol{x}[\sigma_p] = \sum_{j' \leq \sigma_p} \sum_{a' \in \mathcal{A}(j')} \lambda_{(j',a')} \, \boldsymbol{q}_{(j',a')}[\sigma_p] \, \boldsymbol{x}[(j',a')]. \tag{32}$$

However, since $\boldsymbol{q}_{(j',a')} \in Q_{j'}^{(i)}$ for all $j' \leq \sigma_p, a' \in \mathcal{A}(j')$, the sequence-form (probability-mass-conservation) constraints (2) imply that

$$\boldsymbol{q}_{(j',a')}[\sigma_p] = \sum_{a \in \mathcal{A}(j^*)} \boldsymbol{q}_{(j',a')}[(j^*,a)]. \qquad \forall \, j' \leq \sigma_p, a' \in \mathcal{A}(j').$$

Hence, plugging the previous equality into (32), we obtain

$$\left(\sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \leq \sigma_p} \lambda_{\hat{\sigma}} \right) \boldsymbol{x}[\sigma_p] = \sum_{j' \leq \sigma_p} \sum_{a' \in \mathcal{A}(j')} \sum_{a \in \mathcal{A}(j^*)} \lambda_{(j',a')} \, \boldsymbol{q}_{(j',a')}[(j^*,a)] \, \boldsymbol{x}[(j',a')]$$

$$= \sum_{a \in \mathcal{A}(j^*)} \left(\sum_{j' \leq \sigma_p} \sum_{a' \in \mathcal{A}(j')} \lambda_{(j',a')} \, \boldsymbol{q}_{(j',a')}[(j^*,a)] \, \boldsymbol{x}[(j',a')] \right)$$

$$= \sum_{a \in \mathcal{A}(j^*)} \boldsymbol{r}[a],$$

where the last equality follows from recognizing the definition of $\boldsymbol{r}$ on Line 2 of Algorithm 3. So, in conclusion,

$$\sum_{a \in \mathcal{A}(j^*)} \boldsymbol{r}[a] = \left(\sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \leq \sigma_p} \lambda_{\hat{\sigma}} \right) \boldsymbol{x}[\sigma_p]. \tag{33}$$

Part 2: W belongs to $\boldsymbol{x}[\sigma_p] \cdot \mathbb{S}^{\mathcal{A}(j^*)}$. In this second part of the proof, we will prove that all columns of the nonnegative matrix W, defined on Line 3 of Algorithm 3, sum to the same value $\boldsymbol{x}[\sigma_p]$. Fix any $a_c \in \mathcal{A}(j^*)$. Using the definition of W, the sum of the entries in the column of W corresponding

to action a_c is

$$\sum_{a_r \in \mathcal{A}(j^*)} W[a_r, a_c] = \sum_{a_r \in \mathcal{A}(j^*)} r[a_r] + \left(\lambda_{(j^*, a_c)} \boldsymbol{q}_{(j^*, a_c)}[(j^*, a_r)] + \left(1 - \sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \preceq (j^*, a_c)} \lambda_{\hat{\sigma}} \right) \mathbb{1}_{\{a_r = a_c\}} \right) \boldsymbol{x}[\sigma_p]$$

$$= \left(\sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \preceq \sigma_p} \lambda_{\hat{\sigma}} \right) \boldsymbol{x}[\sigma_p] + \boldsymbol{x}[\sigma_p] \lambda_{(j^*, a_c)} \left(\sum_{a_r \in \mathcal{A}(j^*)} \boldsymbol{q}_{(j^*, a_c)}[(j^*, a_r)] \right) + \left(1 - \sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \preceq (j^*, a_c)} \lambda_{\hat{\sigma}} \right) \boldsymbol{x}[\sigma_p]$$

$$= \left(\sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \preceq \sigma_p} \lambda_{\hat{\sigma}} \right) \boldsymbol{x}[\sigma_p] + \boldsymbol{x}[\sigma_p] \lambda_{(j^*, a_c)} + \left(1 - \sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \preceq (j^*, a_c)} \lambda_{\hat{\sigma}} \right) \boldsymbol{x}[\sigma_p],$$

where we used (33) in the second equality, and the fact that $\sum_{a_r \in \mathcal{A}(j^*)} \boldsymbol{q}_{(j^*, a_c)}[(j^*, a_r)] = 1$, since $\boldsymbol{q}_{(j^*, a_c)} \in \mathcal{Q}_{j^*}^{(i)}$ (Definition 2.3), in the third. Using the fact that the set of all predecessors of sequence (j^*, a_c) is the union between all predecessors of the parent sequence σ_p and $\{(j^*, a_c)\}$ itself, after rearranging terms, we can write

$$\sum_{a_r \in \mathcal{A}(j^*)} W[a_r, a_c] = \left(\sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \preceq \sigma_p} \lambda_{\hat{\sigma}} \right) \boldsymbol{x}[\sigma_p] + \boldsymbol{x}[\sigma_p] \lambda_{(j^*, a_c)} + \left(1 - \sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \preceq (j^*, a_c)} \lambda_{\hat{\sigma}} \right) \boldsymbol{x}[\sigma_p]$$

$$= \boldsymbol{x}[\sigma_p] \left(1 + \lambda_{(j^*, a_c)} + \sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \preceq \sigma_p} \lambda_{\hat{\sigma}} - \sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \preceq (j^*, a_c)} \lambda_{\hat{\sigma}} \right)$$

$$= \boldsymbol{x}[\sigma_p].$$

So, all columns of the nonnegative matrix W sum to the same nonnegative quantity $\boldsymbol{x}[\sigma_p]$ and therefore $W \in \boldsymbol{x}[\sigma_p] \cdot \mathbb{S}^{\mathcal{A}(j^*)}$.

Part 3: $\boldsymbol{x}'$ is a $(J \cup \{j^*\})$-partial fixed point of ϕ. We start by arguing that $\boldsymbol{x}'$ satisfies the sequence-form constraints (29) for all $j \in J \cup \{j^*\}$. The crucial observation is that Algorithm 3 only modifies the indices corresponding to sequences (j^*, a) for $a \in \mathcal{A}(j^*)$ and keeps all other entries unmodified. In particular,

$$\boldsymbol{x}'[(j, a)] = \boldsymbol{x}[(j, a)] \qquad \forall \, j \in J, a \in \mathcal{A}(j). \tag{34}$$

Furthermore, because J is a trunk for Player i, the above equation implies that

$$\boldsymbol{x}'[\sigma^{(i)}(j)] = \boldsymbol{x}[\sigma^{(i)}(j)] \qquad \forall \, j \in J.$$

Hence, using the hypothesis that $\boldsymbol{x}$ is a J-partial fixed point of ϕ at the beginning of the call, we immediately conclude that the constraints (29) corresponding to $j \in J$ hold for vector $\boldsymbol{x}'$, and the only condition that remains to be verified is that

$$\boldsymbol{x}'[\sigma_p] = \sum_{a \in \mathcal{A}(j^*)} \boldsymbol{x}'[(j^*, a)]. \tag{35}$$

If $\boldsymbol{x}[\sigma_p] = 0$, then all entries $\boldsymbol{x}'[(j^*, a)]$ are set to 0 (Line 5) and so (35) is trivially satisfied. However, if $\boldsymbol{x}[\sigma_p] \neq 0$, then $\boldsymbol{x}'[(j^*, a)]$ is set to the value $\boldsymbol{x}[\sigma_p] \boldsymbol{b}[a]$ (Line 8), and, since $\boldsymbol{b}$ belongs to the simplex $\Delta^{\mathcal{A}(j^*)}$, (35) holds in this case, too. So, $\boldsymbol{x}'$ satisfies (29) for all $j \in J \cup \{j^*\}$, as we wanted to show.

We now turn our attention to conditions (30). From Lemma 4.13 it follows that $\phi(x)[\sigma]$ only depends on the values of $x[(j', a')]$ for $j' \leq \sigma, a' \in \mathcal{A}(j')$. So, from (34) it follows that

$$\phi(x')[(j, a)] = x[(j, a)] = x'[(j, a)] \qquad \forall\, j \in J, a \in \mathcal{A}(j),$$

and the only condition that remains to be verified is that

$$\phi(x')[(j^*, a^*)] = x'[(j^*, a^*)] \qquad \forall\, a^* \in \mathcal{A}(j^*). \tag{36}$$

Fix any $a^* \in \mathcal{A}(j^*)$. We break the analysis into two cases.

- If $x[\sigma_p] = 0$, then $w = 0$ (Line 5) and therefore $x'[(j^*, a^*)] = 0$. Hence, to show that (36) holds, we need to show that $\phi(x')[(j^*, a^*)] = 0$. To show that, we start from applying Lemma 4.13:

$$\phi(x')[(j^*, a^*)] = \sum_{j' \leq (j^*, a^*)} \sum_{a' \in \mathcal{A}(j')} \lambda_{(j', a')}\, q_{(j', a')}[(j^*, a^*)]\, x'[(j', a')].$$

Now, using the fact that $\{j' \in \mathcal{J}^{(i)} : j' \leq (j^*, a^*)\}$ is equal to the disjoint union $\{j' \in \mathcal{J}^{(i)} : j' \leq \sigma_p\} \cup \{j^*\}$, and that $x'[(j^*, a')] = 0$ for all $a' \in \mathcal{A}(j^*)$, we have

$$\phi(x')[(j^*, a^*)] = \sum_{j' \leq \sigma_p} \sum_{a' \in \mathcal{A}(j')} \lambda_{(j', a')}\, q_{(j', a')}[(j^*, a^*)]\, x'[(j', a')]. \tag{37}$$

Since $q_{(j', a')} \in Q_{j'}^{(i)}$ is a nonnegative vector, from Definition 2.3, it follows that

$$q_{(j', a')}[\sigma_p] = \sum_{a \in \mathcal{A}(j^*)} q_{(j', a')}[(j^*, a)] \geq q_{(j', a')}[(j^*, a^*)]. \tag{38}$$

Hence, substituting (38) into (37),

$$\phi(x')[(j^*, a^*)] \leq \sum_{j' \leq \sigma_p} \sum_{a' \in \mathcal{A}(j')} \lambda_{(j', a')}\, q_{(j', a')}[\sigma_p]\, x'[(j', a')]$$
$$= \phi(x')[\sigma_p] = x'[\sigma_p] = 0,$$

where the first equality follows again from Lemma 4.13, and the second equality follows from the inductive hypothesis that x' is a J-partial fixed point of ϕ. Since x' is a nonnegative vector and ϕ maps nonnegative vectors to nonnegative vectors, we conclude that $\phi(x')[(j^*, a^*)] = 0$, as we wanted to show.

- If $x[\sigma_p] \neq 0$, then b is a fixed point of the stochastic matrix $\frac{1}{x[\sigma_p]}W$, and therefore it satisfies

$$\sum_{a_c \in \mathcal{A}(j^*)} W[a^*, a_c]\, b[a_c] = x[\sigma_p]\, b[a^*].$$

Hence, by using the fact that $x'[(j^*, a^*)] = x[\sigma_p]\, b[a^*]$ (Line 11), we can write

$$x'[(j^*, a^*)] = \sum_{a_c \in \mathcal{A}(j^*)} W[a^*, a_c]\, b[a_c].$$

By expanding the definition of $W[a^*, a_c]$ (Line 3) on the right-hand side

$$x'[(j^*, a^*)] = \sum_{a_c \in \mathcal{A}(j^*)} \left[r[a^*] + \left(\lambda_{(j^*, a_c)} q_{(j^*, a_c)}[(j^*, a^*)] + \left(1 - \sum_{\substack{\hat{\sigma} \in \Sigma_*^{(i)} \\ \hat{\sigma} \preceq (j^*, a_c)}} \lambda_{\hat{\sigma}} \right) \mathbb{1}_{\{a^* = a_c\}} \right) x[\sigma_p] \right] b[a_c]$$

$$= r[a^*] + \left(1 - \sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \preceq (j^*, a^*)} \lambda_{\hat{\sigma}} \right) x'[(j^*, a^*)] + \sum_{a_c \in \mathcal{A}(j^*)} \lambda_{(j^*, a_c)} q_{(j^*, a_c)}[(j^*, a^*)] x'[(j^*, a_c)],$$

where in the second equality we used the fact that $b \in \Delta^{\mathcal{A}(j^*)}$, and the fact that $x'[(j^*, a)] = x[\sigma_p] b[a]$ for all $a \in \mathcal{A}(j^*)$ (Line 11). Expanding the definition of r (Line 2),

$$x'[(j^*, a^*)] = \sum_{j' \preceq \sigma_p} \sum_{a' \in \mathcal{A}(j')} \lambda_{(j', a')} q_{(j', a')}[(j^*, a^*)] x[(j', a')] + \left(1 - \sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \preceq (j^*, a^*)} \lambda_{\hat{\sigma}} \right) x'[(j^*, a^*)]$$

$$+ \sum_{a_c \in \mathcal{A}(j^*)} \lambda_{(j^*, a_c)} q_{(j^*, a_c)}[(j^*, a^*)] x'[(j^*, a_c)]$$

$$= \left(1 - \sum_{\hat{\sigma} \in \Sigma_*^{(i)}, \, \hat{\sigma} \preceq (j^*, a^*)} \lambda_{\hat{\sigma}} \right) x'[(j^*, a^*)] + \sum_{j' \preceq (j^*, a^*)} \sum_{a' \in \mathcal{A}(j')} \lambda_{(j', a')} q_{(j', a')}[(j^*, a^*)] x'[(j', a')]$$

$$= \phi(x')[(j^*, a^*)],$$

where the fact that $\{j' \in \mathcal{J}^{(i)} : j' \preceq (j^*, a^*)\}$ is equal to the disjoint union $\{j' \in \mathcal{J}^{(i)} : j' \preceq \sigma_p\} \cup \{j^*\}$ in the second equality, and Lemma 4.13 in the third equality.

Part 4: Complexity analysis. In this part, we bound the number of operations required by Algorithm 3.

- Line 2: Each entry $r[a]$ can be trivially computed in $O(|\Sigma^{(i)}|)$ time by traversing all predecessors of j^*. So, the vector r requires $O(|\Sigma^{(i)}| |\mathcal{A}(j^*)|)$ operations to be computed.
- Line 3: If $a_r = a_c$, then the number of operations required to compute $W[a_r, a_c]$ is dominated by the computation of $\sum_{\hat{\sigma} \preceq (j^*, a_c)} \lambda_{\hat{\sigma}}$, which requires $O(|\Sigma^{(i)}|)$ operations. Otherwise, if $a_r \ne a_c$, then the computation of $W[a_r, a_c]$ can be carried out in a constant number of operations. Hence, the computation of $W[a_r, a_c]$ for all $a_r, a_c \in \mathcal{A}(j^*)$ requires $O(|\Sigma^{(i)}| |\mathcal{A}(j^*)| + |\mathcal{A}(j^*)|^2)$ time. Since $|\mathcal{A}(j^*)| \le |\Sigma^{(i)}|$, the total number of operations required to compute all entries of W is $O(|\Sigma^{(i)}| |\mathcal{A}(j^*)|)$.
- Lines 4 to 8: If $x[\sigma_p] = 0$, then the computation of w requires $O(|\mathcal{A}(j^*)|)$ operations. If, however, $x[\sigma_p] \ne 0$, then the computation of w requires $O(\mathsf{FP}(|\mathcal{A}(j^*)|) + |\mathcal{A}(j^*)|)$ operation. Since clearly any fixed point oracle for a square matrix of order $|\mathcal{A}(j^*)|$ needs to spend time at least $\Omega(|\mathcal{A}(j^*)|)$ time writing the output, $O(\mathsf{FP}(|\mathcal{A}(j^*)|) + |\mathcal{A}(j^*)|) = O(\mathsf{FP}(|\mathcal{A}(j^*)|))$. So, no matter the value of $x[\sigma_p]$, the number of iterations is bounded by $O(\mathsf{FP}(|\mathcal{A}(j^*)|))$.
- Line 11: finally, the algorithm spends $O(|\mathcal{A}(j^*)|)$ operations to set entries of x.

Summing the number of operations of each of the different steps of the algorithm, we conclude that each call to EXTEND(ϕ, J, j^*, x) requires at most $O(|\Sigma^{(i)}| |\mathcal{A}(j^*)| + \mathsf{FP}(|\mathcal{A}(j^*)|))$ operations. □

A fixed point for $\phi \in \mathrm{co}(\Psi^{(i)})$ can be computed by repeatedly invoking EXTEND to grow the trunk J one information set at a time, until $J = \mathcal{J}^{(i)}$, starting from the $\emptyset$-partial fixed point

---

ALGORITHM 4: FIXEDPOINT(ϕ)

Input : • $\phi = \sum_{\hat{\sigma} \in \Sigma_*^{(i)}} \lambda_{\hat{\sigma}} \phi_{\hat{\sigma} \to q_{\hat{\sigma}}}^{(i)} \in \text{co}(\Psi^{(i)})$ transformation for a player $i \in [n]$

Output: • $q \in Q^{(i)}$ such that $q = \phi(q)$

1   $q \leftarrow 0 \in \mathbb{R}^{\Sigma^{(i)}}$, $q[\varnothing] \leftarrow 1$

2   $J \leftarrow \emptyset$

3   **for** $j \in \mathcal{J}^{(i)}$ *in top-down order*[13] **do**

4      $q \leftarrow$ EXTEND(ϕ, J, j, q)   [▷ Algorithm 3]

5      $J \leftarrow J \cup \{j\}$

6   **return** q

$x_0 \in \mathbb{R}_{\geq 0}^{\Sigma^{(i)}}$ introduced in Lemma 4.11. This leads to Algorithm 4, whose correctness and computational complexity is a straightforward corollary of Proposition 4.14.

COROLLARY 4.15. *Let $i \in [n]$ be a player, and let $\phi = \sum_{\hat{\sigma} \in \Sigma_*^{(i)}} \lambda_{\hat{\sigma}} \phi_{\hat{\sigma} \to q_{\hat{\sigma}}}^{(i)}$ be a transformation in* $\text{co}(\Psi^{(i)})$ *expressed as in (28). Then, Algorithm 4 computes a fixed point $Q^{(i)} \ni q = \phi(q)$ in time upper bounded as $O(|\Sigma^{(i)}|^2 + \sum_{j \in \mathcal{J}^{(i)}} \text{FP}(|\mathcal{A}(j)|))$.*

ALGORITHM 5: $\text{co}(\Psi^{(i)})$-regret minimizer $\bar{\mathcal{R}}^{(i)}$ for the set $Q^{(i)}$

Data: • $i \in [n]$ player

        • $\tilde{\mathcal{R}}^{(i)}$ regret minimizer for $\Psi^{(i)}$, defined in Algorithm 2

1   **function** NEXTELEMENT()

2      $\phi^t = \sum_{\hat{\sigma} \in \Sigma_*^{(i)}} \lambda_{\hat{\sigma}}^t \phi_{\hat{\sigma} \to q_{\hat{\sigma}}^t}^{(i)} \in \text{co}(\Psi^{(i)}) \leftarrow \tilde{\mathcal{R}}^{(i)}.\text{NEXTELEMENT}()$ [▷ Algorithm 2]

3      $q^t \in Q^{(i)} \leftarrow$ FIXEDPOINT(ϕ^t)   [▷ Algorithm 4]

4   **return** q^t

5   **function** OBSERVEUTILITY(ℓ^t)

6      $L^t \leftarrow$ linear utility function $\phi \mapsto \ell^t(\phi(q^t))$

7      $\tilde{\mathcal{R}}^{(i)}.\text{OBSERVEUTILITY}(L^t)$ [▷ Algorithm 2]

4.5 The Complete Algorithm

In this subsection, we put together all the pieces we constructed in the previous subsections to build a $\Psi^{(i)}$-regret minimizer that satisfies all requirements of Problem 1.

First, we provide in Algorithm 5 our $\text{co}(\Psi^{(i)})$-regret minimizer. Its correctness follows from the correctness of the construction by Gordon et al. [20], described in Section 2.4, and by using Theorem 4.6 and Corollary 4.15.

THEOREM 4.16. *Let $i \in [n]$ be any player. $\bar{\mathcal{R}}^{(i)}$, defined in Algorithm 5, is a $\text{co}(\Psi^{(i)})$-regret minimizer for the set of sequence-form strategies $Q^{(i)}$, whose cumulative regret upon observing linear utility functions $\ell^1, \ldots, \ell^T$ satisfies*

$$R^T \leq 2D|\Sigma^{(i)}|\sqrt{T},$$

where D is any constant such that $\max_{q,q'}\{\ell^t(q) - \ell^t(q')\} \leq D$ for all $t = 1, \ldots, T$. Furthermore, the OBSERVEUTILITY operation requires time $O(|\Sigma^{(i)}|^2)$, and the NEXTELEMENT operation requires time $O(|\Sigma^{(i)}|^2 + \sum_{j \in \mathcal{J}^{(i)}} \text{FP}(|\mathcal{A}(j)|))$ at all t.

---

[13]That is, according to a pre-order tree traversal: If $j \prec j'$, then j appears before j' in the iteration order.

ALGORITHM 6: $\Psi^{(i)}$-regret minimizer $\mathcal{R}^{(i)}$ for the set $\Pi^{(i)}$

Data: • $i \in [n]$ player
   • $\bar{\mathcal{R}}^{(i)}$, co$(\Psi^{(i)})$-regret minimizer for $Q^{(i)}$, defined in Algorithm 5

1 **function** NextElement()
2 | $q^t \in Q^{(i)} \leftarrow \bar{\mathcal{R}}^{(i)}$.NextElement()   [▷ Algorithm 5]
3 | Sample[14] a deterministic sequence-form strategy $\pi^t \in \Pi^{(i)}$ so it is an unbiased estimator of q^t,
   | using the natural sampling scheme described in Section 2.2
4 | **return** π^t

5 **function** ObserveUtility(ℓ^t)
6 | $\bar{\mathcal{R}}^{(i)}$.ObserveUtility(ℓ^t)   [▷ Algorithm 5]

PROOF. From the properties of Gordon et al.'s construction [20] (Section 2.4), the cumulative $\Psi^{(i)}$-regret incurred by $\bar{\mathcal{R}}^{(i)}$ is equal, at all times, to the cumulative regret incurred by the underlying regret minimizer $\tilde{\mathcal{R}}^{(i)}$ for the set of deviations $\Psi^{(i)}$. So, the regret bound follows from the regret analysis of Theorem 4.6.

Similarly, the complexity analysis follows from combining the analysis of $\tilde{\mathcal{R}}^{(i)}$ and of FIXED-POINT (Algorithm 4), together with the observation that the canonical representation $\langle L^t \rangle$ of the linear utility function co$(\Psi^{(i)}) \ni \phi \mapsto \ell^t(\phi(q^t))$ is the matrix $\langle \ell^t \rangle (q^t)^\top$, which can be trivially computed in $O(|\Sigma^{(i)}|^2)$ time.   □

Since co$(\Psi^{(i)}) \supseteq \Psi^{(i)}$, Algorithm 5 is in particular also a $\Psi^{(i)}$-regret minimizer for the set $Q^{(i)}$, and thus Theorem 4.16 establishes that Algorithm 5 provides a solution to Problem 2.

To obtain our $\Psi^{(i)}$-regret minimizer for the set of pure sequence-form strategies $\Pi^{(i)}$ from Algorithm 5, we apply the construction described in Section 4.2. The resulting regret minimizer, $\mathcal{R}^{(i)}$, is given in Algorithm 6. Applying Lemma 4.1 immediately yields the following corollary:

COROLLARY 4.17. *Let $i \in [n]$ be any player. $\mathcal{R}^{(i)}$, defined in Algorithm 6, is a $\Psi^{(i)}$-regret minimizer for the set $\Pi^{(i)}$, whose cumulative regret R^T upon observing linear utility functions $\ell^1, \ldots, \ell^T$ satisfies*

$$R^T \leq 2D|\Sigma^{(i)}|\sqrt{T} + 4D\sqrt{T|\Sigma^{(i)}|\log(1/\delta)} \text{ with probability at least } 1 - \delta,$$

for any $\delta \in (0, 1)$, where D is any constant such that $\max_{q, q'}\{\ell^t(q) - \ell^t(q')\} \leq D$ for all $t = 1, \ldots, T$. Furthermore, the OBSERVEUTILITY operation runs in $O(|\Sigma^{(i)}|^2)$ time, and the NEXTELEMENT operation runs in $O(|\Sigma^{(i)}|^2 + \sum_{j \in \mathcal{J}^{(i)}} \mathsf{FP}(|\mathcal{A}(j)|))$ time at all t.

Therefore, Algorithm 6 provides a regret minimizer that satisfies all requirements of Problem 1.

5   CONVERGENCE TO EFCE

Theorem 3.8 implies that if all players $i \in [n]$ play the game repeatedly according to the outputs of a $\Psi^{(i)}$-regret minimizer for $\Pi^{(i)}$ that observes, at each time t, the linear utility function given in (10), then the empirical frequency of play is a $(\frac{1}{T} \max_i R^{(i), T})$-EFCE, where $R^{(i), T}$ is the regret cumulated by the $\Psi^{(i)}$-regret minimizer for Player i.

In particular, when all players play according to the strategies recommended by Algorithm 6, the following can be shown by combining Theorem 4.17 and Theorem 3.8:

---

[14]As discussed in Lemma 4.1, in principle, any unbiased sampling scheme will work. For the purposes of analyzing the complexity of Algorithm 6, however, we will assume that the natural sampling scheme described in Section 2.2 is used. That sampling scheme runs in linear time in $|\Sigma^{(i)}|$.

THEOREM 5.1. *When all players $i = 1, \ldots, n$ play according to the outputs of the regret minimizer $\mathcal{R}^{(i)}$ defined in Algorithm 6, receiving as feedback at all times t the linear utility functions $\ell^{(i), t}$ defined in (10), the empirical frequency of play after T repetitions of the game is a*

$$\left(D\frac{2|\mathcal{H}| + 4\sqrt{|\mathcal{H}| \log(n/\delta)}}{\sqrt{T}}\right)\text{-EFCE with probability at least } 1 - \delta,$$

for any $\delta \in (0, 1)$, where D is the difference between the maximum and minimum payoff of the game, and $|\mathcal{H}|$ is the number of nodes in the game tree.

PROOF. Let $R^{(i), T}$ be the regret cumulated by $\mathcal{R}^{(i)}$ (Algorithm 6) up to time T. From Theorem 4.17, we have that for all $\delta' \in (0, 1)$,

$$\mathbb{P}\left[R^{(i), T} \le 2D|\mathcal{H}|\sqrt{T} + 4D\sqrt{T|\mathcal{H}| \log(1/\delta')}\right] \ge \mathbb{P}\left[R^{(i), T} \le 2D|\Sigma^{(i)}|\sqrt{T} + 4D\sqrt{T|\Sigma^{(i)}| \log(1/\delta')}\right]$$

$$\ge 1 - \delta',$$

where the first inequality follows from the fact that $|\Sigma^{(i)}| = \sum_{j \in \mathcal{J}^{(i)}} |\mathcal{A}(j)| \le \sum_{h \in \mathcal{H}} |\mathcal{A}(h)| \le |\mathcal{H}|$ (the number of edges in a tree is always less than the number of nodes). So,

$$\mathbb{P}\left[\max_i R^{(i), T} \le 2D|\mathcal{H}|\sqrt{T} + 4D\sqrt{T|\mathcal{H}| \log(1/\delta')}\right]$$

$$= \mathbb{P}\left[\bigcap_i \left\{R^{(i), T} \le 2D|\mathcal{H}|\sqrt{T} + 4D\sqrt{T|\mathcal{H}| \log(1/\delta')}\right\}\right]$$

$$\ge 1 - n\delta',$$

where the inequality follows from the union bound. Substituting $\delta := n\delta'$ and using Theorem 3.8 yields the result. □

A standard application of the Borel-Cantelli lemma enables us to move from the high-probability guarantees at finite time of Theorem 5.1 to almost-sure guarantees in the limit.

COROLLARY 5.2. *When all players $i = 1, \ldots, n$ play infinitely many repetitions of the game according to the outputs of the regret minimizer $\mathcal{R}^{(i)}$ defined in Algorithm 6, receiving as feedback at all times t the linear utility functions $\ell^{(i), t}$ defined in (10), the empirical frequency of play converges, almost surely, to the set of EFCEs.*

ACKNOWLEDGMENTS

REFERENCES

[1] Robert J. Aumann. 1974. Subjectivity and correlation in randomized strategies. *J. Math. Econ.* 1, 1 (1974), 67–96.
[2] Kazuoki Azuma. 1967. Weighted sums of certain dependent random variables. *Tohoku Math. J.* 19, 3 (1967), 357–367.
[3] Noam Brown and Tuomas Sandholm. 2018. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science* 359, 6374 (2018), 418–424.
[4] Noam Brown and Tuomas Sandholm. 2019. Solving imperfect-information games via discounted regret minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1829–1836.
[5] Amotz Cahn. 2004. General procedures leading to correlated equilibria. *Int. J. Game Theor.* 33, 1 (2004), 21–40.

[6] A. Celli and N. Gatti. 2018. Computational results for extensive-form adversarial team games. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

[7] Andrea Celli, Alberto Marchesi, Tommaso Bianchi, and Nicola Gatti. 2019. Learning to correlate in multi-player general-sum sequential games. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*. 13055–13065.

[8] Andrea Celli, Alberto Marchesi, Gabriele Farina, and Nicola Gatti. 2020. No-regret learning dynamics for extensive-form correlated equilibrium. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*.

[9] Nicolo Cesa-Bianchi and Gábor Lugosi. 2006. *Prediction, Learning, and Games*. Cambridge University Press.

[10] Xi Chen and Xiaotie Deng. 2006. Settling the complexity of two-player Nash equilibrium. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 261–272.

[11] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. 2009. The complexity of computing a Nash equilibrium. *SIAM J. Comput.* 39, 1 (2009), 195–259.

[12] Constantinos Daskalakis and Christos H. Papadimitriou. 2009. On a network generalization of the minmax theorem. In *Proceedings of the International Colloquium on Automata, Languages, and Programming*. Springer, 423–434.

[13] Miroslav Dudík and Geoffrey J. Gordon. 2009. A sampling-based approach to computing equilibria in succinct extensive-form games. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. 151–160.

[14] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. 2019. Regret circuits: Composability of regret minimizers. In *Proceedings of the International Conference on Machine Learning*. 1863–1872.

[15] Gabriele Farina, Chun Kai Ling, Fei Fang, and Tuomas Sandholm. 2019. Correlation in extensive-form games: Saddle-point formulation and benchmarks. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*. 9229–9239.

[16] Dean P. Foster and Rakesh V. Vohra. 1997. Calibrated learning and correlated equilibrium. *Games Econ. Behav.* 21, 1-2 (1997), 40.

[17] Drew Fudenberg and David K. Levine. 1995. Consistency and cautious fictitious play. *J. Econ. Dynam. Contr.* 19, 5-7 (1995), 1065–1089.

[18] Drew Fudenberg and David K. Levine. 1998. *The Theory of Learning in Games*, Vol. 2. MIT Press.

[19] Drew Fudenberg and David K. Levine. 1999. Conditional universal consistency. *Games Econ. Behav.* 29, 1-2 (1999), 104–130.

[20] Geoffrey J. Gordon, Amy Greenwald, and Casey Marks. 2008. No-regret learning in convex games. In *Proceedings of the International Conference on Machine Learning*. 360–367.

[21] Amy Greenwald and Amir Jafari. 2003. A general class of no-regret learning algorithms and game-theoretic equilibria. In *Learning Theory and Kernel Machines*. Springer, 2–12.

[22] Sergiu Hart and Andreu Mas-Colell. 2000. A simple adaptive procedure leading to correlated equilibrium. *Econometrica* 68, 5 (2000), 1127–1150.

[23] Sergiu Hart and Andreu Mas-Colell. 2001. A general class of adaptive strategies. *J. Econ. Theor.* 98, 1 (2001), 26–54.

[24] Sergiu Hart and Andreu Mas-Colell. 2003. Uncoupled dynamics do not lead to Nash equilibrium. *Amer. Econ. Rev.* 93, 5 (2003), 1830–1836.

[25] Elad Hazan and Satyen Kale. 2008. Computational equivalence of fixed points and no regret algorithms, and convergence to equilibria. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*.

[26] Wassily Hoeffding. 1963. Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.* 58, 301 (1963), 13–30.

[27] Wan Huang and Bernhard von Stengel. 2008. Computing an extensive-form correlated equilibrium in polynomial time. In *Proceedings of the International Workshop on Internet and Network Economics*. Springer, 506–513.

[28] Albert Xin Jiang and Kevin Leyton-Brown. 2015. Polynomial-time computation of exact correlated equilibrium in compact games. *Games Econ. Behav.* 91 (2015), 347–359.

[29] Sham Kakade, Michael Kearns, John Langford, and Luis Ortiz. 2003. Correlated equilibria in graphical games. In *Proceedings of the 4th ACM Conference on Electronic Commerce*. 42–47.

[30] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. 1996. Efficient computation of equilibria for extensive two-person games. *Games Econ. Behav.* 14, 2 (1996), 247–259.

[31] Elias Koutsoupias and Christos Papadimitriou. 1999. Worst-case equilibria. In *Proceedings of the Annual Symposium on Theoretical Aspects of Computer Science*. Springer, 404–413.

[32] H. W. Kuhn. 1953. *Extensive Games and the Problem of Information*. Princeton University Press, 193–216.

[33] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael H. Bowling. 2009. Monte Carlo sampling for regret minimization in extensive games. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*. 1078–1086.

[34] Colin McDiarmid. 1998. *Concentration*. Springer, Berlin, 195–248.

[35] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. 2017. DeepStack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* 356, 6337 (2017), 508–513.

[36] Dustin Morrill, Ryan D'Orazio, Marc Lanctot, James R. Wright, Michael Bowling, and Amy R. Greenwald. 2021. Efficient deviation types and learning for hindsight rationality in extensive-form games. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research,* Vol. 139*)*. PMLR, 7818–7828.

[37] Dustin Morrill, Ryan D'Orazio, Reca Sarfati, Marc Lanctot, James Wright, Amy Greenwald, and Michael Bowling. 2020. Hindsight and sequential rationality of correlated play. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI'20)*.

[38] John F. Nash. 1950. Equilibrium points in n-person games. *Proc. Nat. Acad. Sci.* 36, 1 (1950), 48–49.

[39] C. C. Paige, George P. H. Styan, and Peter G. Wachter. 1975. Computation of the stationary distribution of a Markov chain. *J. Statist. Computat. Simul.* 4, 3 (1975), 173–186.

[40] Christos H. Papadimitriou and Tim Roughgarden. 2008. Computing correlated equilibria in multi-player games. *J. ACM* 55, 3 (2008), 14.

[41] I. Romanovskii. 1962. Reduction of a game with complete memory to a matrix game. *Soviet Math.* 3 (1962).

[42] Tim Roughgarden and Éva Tardos. 2002. How bad is selfish routing? *J. ACM* 49, 2 (2002), 236–259.

[43] Yoav Shoham and Kevin Leyton-Brown. 2008. *Multiagent Systems: Algorithmic, Game-theoretic, and Logical Foundations*. Cambridge University Press.

[44] Gilles Stoltz and Gábor Lugosi. 2007. Learning correlated equilibria in games with compact sets of strategies. *Games Econ. Behav.* 59, 1 (2007), 187–208.

[45] Oskari Tammelin. 2014. Solving large imperfect information games using CFR+. *arXiv preprint arXiv:1407.5042* (2014).

[46] Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. 2015. Solving heads-up limit Texas hold'em. In *Proceedings of the International Joint Conferences on Artificial Intelligence*. 645–652.

[47] Bernhard von Stengel. 1996. Efficient computation of behavior strategies. *Games Econ. Behav.* 14, 2 (1996), 220–246.

[48] Bernhard von Stengel and Françoise Forges. 2008. Extensive-form correlated equilibrium: Definition and computational complexity. *Math. Oper. Res.* 33, 4 (2008), 1002–1022.

[49] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. 2008. Regret minimization in games with incomplete information. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*. 1729–1736.