15-888 Computational Game Solving (Fall 2025) Monday, September 15th, 2025

Lecture 6
Learning in General-Sum Games: Correlated Equilibria and Φ-Regret

Ioannis Anagnostides

Having introduced efficient algorithms for minimizing regret in an online environment, we start focusing on what happens when all players in a multi-player *general-sum* game employ no-regret learning to update their strategies.

To answer this question, we begin by introducing and motivating the concept of *(coarse) correlated equilibrium* (Section 1); this is a relaxation of the Nash equilibrium in which players no longer have to randomize independently. We will see that correlated equilibria enjoy more favorable computational properties than their uncorrelated counterparts. To connect correlated equilibria with regret minimization, we will introduce the notion of Φ*-regret* (Section 1.2), which strengthens the usual notion of (external) regret covered in an earlier lecture. The upshot is that the distribution of play arising from players minimizing Φ-regret converges to the set of correlated equilibria. We will then present, in Section 2, a general, abstract framework for minimizing Φ-regret, and will then see how this framework can be applied to minimize *swap regret*—and thereby converge to correlated equilibria—in normal-form games.

1 Correlated and coarse correlated equilibria

A major criticism of the Nash equilibrium is that, even though one always exists [Nash, 1950], it is computationally intractable to find one [Daskalakis et al., 2008]—let alone an optimal one [Gilboa and Zemel, 1989]. In light of this fact, we shouldn't expect simple learning algorithms—such as online gradient descent or regret matching—to converge to Nash equilibria; but, then, what are no-regret dynamics converging to in general-sum games? As we mentioned in passing in a previous lecture, no-regret learning is inherently tied to the notion of *coarse correlated equilibrium* [Moulin and Vial, 1978]. Let's begin by recalling the basic definition and start building some intuition about this solution concept; we confine our discussion to normal-form games.

Definition 1.1 (Coarse correlated equilibrium)**.** A correlated distribution $\boldsymbol{\mu} \in \Delta(\mathcal{A}_1 \times \cdots \times \mathcal{A}_n)$ is an ϵ*-coarse correlated equilibrium (CCE)* if for any player $i \in [n]$ and deviation $a_i' \in \mathcal{A}_i$,

$$\mathbb{E}_{(a_1,\dots,a_n)\sim\mu} u_i(a_1,\dots,a_n) \geq \mathbb{E}_{(a_1,\dots,a_n)\sim\mu} u_i(a_i', a_{-i}) - \epsilon.$$

This definition echoes the Nash equilibrium notion, but with one crucial difference: the underlying distribution $\boldsymbol{\mu}$ in Definition 1.1 can be *correlated*, while in a Nash equilibrium $\boldsymbol{\mu}$ has to be a *product distribution*—reflecting the fact that players randomize independently. To explain this difference, let's consider the following two distributions with respect to some 2×2 bimatrix game (meaning that each of the two players has two available actions).

$$\boldsymbol{\mu} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}, \quad \boldsymbol{\mu}' = \begin{bmatrix} 1/6 & 1/6 \\ 1/3 & 1/3 \end{bmatrix}.$$

Both are distributions over $\mathcal{A}_1 \times \mathcal{A}_2 = \{1, 2\} \times \{1, 2\}$, but we claim that only $\boldsymbol{\mu}'$ is a product distribution. Indeed, if Player 1—the row player—plays $(1/3, 2/3)$ and Player 2 plays $(1/2, 1/2)$, the induced distribution over the 4 outcomes of the game exactly matches $\boldsymbol{\mu}'$. By contrast, it's easy to see that no pair of strategies gives rise to $\boldsymbol{\mu}$.

It follows readily from Definition 1.1 that a Nash equilibrium is always a CCE; a Nash equilibrium is basically an *uncorrelated* (coarse) correlated equilibrium. But the set of CCEs can unlock new, sometimes more desirable, outcomes. Before we examine a concrete example, let's also introduce the stronger notion of a *correlated equilibrium*, due to Aumann [1974].

Definition 1.2 (Correlated equilibrium)**.** A correlated distribution $\boldsymbol{\mu} \in \Delta(\mathcal{A}_1 \times \cdots \times \mathcal{A}_n)$ is an *ϵ-correlated equilibrium (CE)* if for any player $i \in [n]$ and deviation function $\phi_i : \mathcal{A}_i \to \mathcal{A}_i$,

$$\mathbb{E}_{(a_1,\ldots,a_n)\sim\mu} u_i(a_1, \ldots, a_n) \geq \mathbb{E}_{(a_1,\ldots,a_n)\sim\mu} u_i(\phi_i(a_i), a_{-i}) - \epsilon.$$

Definitions 1.1 and 1.2 can both be interpreted through the use of a trusted third party—a *mediator* or *correlation device*—who samples an action profile $(a_1, \ldots, a_n)$ from the correlated distribution $\boldsymbol{\mu}$, and then provides a_i to each player $i \in [n]$ as a *recommendation.* A distribution is a CCE or a CE if no player has an incentive to deviate from the recommendation, but for CEs the set of possible deviations is richer, making that set of equilibria tighter. In particular, in a CE, a player can decide whether to deviate *after* observing the recommendation, while in a CCE the decision has to be made beforehand. From that perspective, a CCE may seem harder to justify, for one would need some binding mechanism to ensure the player would not be able to deviate after observing the recommendation.

Example 1.3. Let's look at a concrete example to explain these concepts. We consider the "game of chicken." This is a 2×2 game—played between two competitive drivers who are fast approaching an intersection from different streets—whose utilities are tabulated in Figure 1. Each player can either play Stop or Go. If both players elect Go a crash ensues, a bad outcome for both players. If a player stalls, it gets no utility from the game, while if it proceeds while the other player stops, it gets a utility of 1 for managing to quickly cross the intersection without crashing.

It's easy to see that this game has exactly three Nash equilibria: i) (Go, Stop), ii) (Stop, Go), and $((5/6, 1/6), (5/6, 1/6))$, meaning that both players play Stop with probability $5/6$. From these three outcomes, the first two are *not* equitable in that they favor one player over the other. The third outcome is even worse: it leads to a crash with some non-negligible probability, so much so that each player gets zero utility in expectation.

Fortunately, (C)CEs address these issues. In particular, let's consider the correlated distribution $\frac{1}{2}(\text{Go, Stop}) + \frac{1}{2}(\text{Stop, Go})$. It's easy to verify that this is a CE, and thus a CCE as well. Under that outcome, both players get in expectation a utility of $1/2$. Turning our attention to CEs, there is a natural interpretation of this outcome through a *traffic light*, which provides a signal to each player. If Player 1 is recommended Stop, it means that Player 2 will play Go with probability 1, so stopping is in Player 1's interest. On the flip side, if Player 1 is recommended Go, it means

that Player 2 will play Stop with probability 1, so proceeding is safe for Player 1. In other words, in a CE, the signal a player observes updates that player's beliefs concerning the behavior of the other players. (A Nash equilibrium can be seen as the corner case in which the signal carries no pertinent information on account of the fact that players are acting independently.)

	Stop	Go
Stop	0, 0	0, 1
Go	1, 0	-5, -5

Figure 1: The game of chicken.

Example 1.4. The purpose of our next example is to further clarify the difference between CCEs and CEs. Let's consider a 4×4 bimatrix game described with the payoff matrices

$$\mathbf{R} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{bmatrix} \text{ and } \mathbf{C} = \begin{bmatrix} 2 & 0 & 3 & 0 \\ 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{1}$$

for the row and column player, respectively. We label each player's actions as 1, 2, 3, and 4. We claim that the distribution $\boldsymbol{\mu} = \frac{1}{2}(1, 1) + \frac{1}{2}(2, 2)$ is an exact CCE of game (1), whereas the CE gap of $\boldsymbol{\mu}$ is large—namely, 1. The swap deviation ϕ that results in a large deviation gain is such that $1 \mapsto 3$ and $2 \mapsto 4$; the mapping for the rest of the actions is moot, as they are almost surely never played under $\boldsymbol{\mu}$. While each player obtains a utility of 2 under $\boldsymbol{\mu}$, deviating according to ϕ gives a utility of 3. Even so, it can be verified that $\boldsymbol{\mu}$ is a CCE as it is robust with respect to constant deviations.

1.1 Computational properties

We have seen that CCEs and CEs unlock new outcomes not attainable under independent randomization. What's more, they have better computational properties than Nash equilibria. Specifically, the set of (C)CEs is convex and can be described through a linear program.

Proposition 1.5. *There is a linear program with $\prod_{i=1}^{n} |\mathcal{A}_i|$ variables and $\sum_{i=1}^{n} |\mathcal{A}_i|(|\mathcal{A}_i| - 1)$ constraints whose solution is an exact correlated equilibrium of the game.*

While the number of swap deviations of each player $i \in [n]$ is $|\mathcal{A}_i|^{|\mathcal{A}_i|}$, to arrive at Proposition 1.5, it's enough to consider only a certain subset of swap deviations—ones that only change a *single* action; such a deviation is called *internal*—with size $|\mathcal{A}_i|(|\mathcal{A}_i| - 1)$; the simple proof is left as an exercise.

The key caveat with Proposition 1.5 is that the size of the LP grows exponentially with the number of players; the basic reason why this happens is that a correlated distribution in multiplayer games is an exponential objective—one needs to specify the value of $\prod_{i=1}^{n} |\mathcal{A}_i| - 1$ coordinates. If time permits, next lecture will cover a sophisticated, famous algorithm for addressing

this issue; for now, at least we know that we can compute a CE in games with a constant number of players. Furthermore, one can also incorporate any linear objective function into the linear program, such as the *social welfare*—the sum of the players' utilities.

1.2 Connection to no-regret learning

As we have alluded to, (coarse) correlated equilibria are closely tied to the framework of regret minimization in online learning.

Φ-regret To formalize this connection in its full generality, we will now introduce the important concept of Φ-*regret*. It is a measure of the learner's performance parameterized by a family of *strategy deviations* Φ. In what follows, we operate in the usual online learning setting: in every round $t \in [T]$ the learner first specifies a strategy $\boldsymbol{x}^{(t)} \in \mathcal{X}$ that lies in a convex and compact set $\mathcal{X}$, whereupon the environment devises a linear utility function $\boldsymbol{x} \mapsto \langle \boldsymbol{x}, \boldsymbol{u}^{(t)} \rangle$ for some utility vector $\boldsymbol{u}^{(t)}$. For a set of deviations Φ comprising functions $\phi : \mathcal{X} \to \mathcal{X}$, Φ-regret is defined as

$$\Phi\text{Reg}^{(T)} \coloneqq \max_{\phi \in \Phi} \left\{ \sum_{t=1}^{T} \langle \phi(\boldsymbol{x}^{(t)}), \boldsymbol{u}^{(t)} \rangle \right\} - \sum_{t=1}^{T} \langle \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \rangle. \tag{2}$$

An earlier lecture focused on the special case where Φ comprises only *constant deviations*: $\Phi_{\text{const}} = \{\phi : \exists \boldsymbol{x}' \in \mathcal{X} \text{ such that } \phi(\boldsymbol{x}) = \boldsymbol{x}'\}$; this is the most standard definition of regret in online learning, typically referred to as *external* regret to disambiguate with other stronger notions. The key point about (2) is that the richer the set of deviations Φ, the stronger the induced notion of hindsight rationality. The other end of the spectrum where Φ consists of all possible deviations $\mathcal{X} \to \mathcal{X}$ is called *swap regret*. As expected, an algorithm can experience large swap regret even when its external regret is small.

Example 1.6. Let's say the learner picks a distribution over three actions, 1, 2, and 3. Suppose further that the sequence of utilities and selected actions follow the pattern of Figure 2, where $T = 0 \mod 3$. We see that the learner collects overall a utility of $T/3$, which in fact matches the optimal strategy in hindsight. So, the external regret of the learner is 0 in this example. On the other hand, consider the swap deviation

$$\phi : a \mapsto \begin{cases} 2 & \text{if } a = 1, \\ 1 & \text{if } a = 2 \\ 3 & \text{if } a = 3. \end{cases}$$

Under that deviation, the learner would be able to collect maximal utility, meaning that the swap regret of the learner is $2T/3 = \Omega(T)$.

The techniques we have covered so far for minimizing external regret will not be enough to minimize swap regret; for example, it is known that multiplicative weights update (MWU) can have linear swap regret [Cesa-Bianchi and Lugosi, 2006]. Before we introduce some new algorithmic ideas to cope with the richer set of deviations (Section 2), let's first formalize the connection

1	0	0	1	0	0	1	0	0	1
2	0	1	0	0	1	0	0	1	0
3	1	0	0	1	0	0	1	0	0

Figure 2: An example of a learner with large swap regret but zero external regret.

between minimizing Φ-regret and correlated equilibrium concepts. In accordance with (2), we expand the scope of Definition 1.2 to general *multilinear games*. Here, each player $i \in [n]$ selects a strategy $\boldsymbol{x}_i \in \mathcal{X}_i$ from a convex and compact set $\mathcal{X}_i$, so that for any joint strategy $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$, the utility can be expressed as $u_i(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \langle \boldsymbol{x}_i, \boldsymbol{u}_i(\boldsymbol{x}_{-i}) \rangle$ for some utility vector $\boldsymbol{u}_i$ that does not depend on $\boldsymbol{x}_i$. This is a useful abstraction for encompassing both normal- and extensive-form games, the latter under the sequence-form representation covered in the previous lecture.

Definition 1.7 (Φ-equilibrium)**.** A correlated distribution $\boldsymbol{\mu} \in \Delta(\mathcal{X}_1 \times \cdots \times \mathcal{X}_n)$ is an ϵ-Φ-*equilibrium* if for any player $i \in [n]$ and deviation function $\Phi_i \ni \phi_i : \mathcal{X}_i \to \mathcal{X}_i$,

$$\mathbb{E}_{(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)\sim\mu} u_i(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \geq \mathbb{E}_{(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)\sim\mu} u_i(\phi_i(\boldsymbol{x}_i), \boldsymbol{x}_{-i}) - \epsilon.$$

Theorem 1.8. *Suppose that each player $i \in [n]$ incurs Φ_i-regret $\Phi\mathrm{Reg}_i^{(T)}$ under the sequence of utilities $(\boldsymbol{u}_i(\boldsymbol{x}_{-i}^{(t)}))_{t=1}^T$. Then the average correlated distribution of play $\boldsymbol{\mu} \coloneqq \frac{1}{T}\sum_{t=1}^T \boldsymbol{x}_1^{(t)} \otimes \cdots \otimes \boldsymbol{x}_n^{(t)}$ is an ϵ-Φ-equilibrium with $\epsilon = \frac{1}{T}\max_{1\leq i\leq n} \Phi\mathrm{Reg}_i^{(T)}$.*

Above, $\boldsymbol{x}_1^{(t)} \otimes \cdots \otimes \boldsymbol{x}_n^{(t)}$ is the product distribution induced by $(\boldsymbol{x}_1^{(t)}, \dots, \boldsymbol{x}_n^{(t)})$; that is, $\otimes$ denotes the tensor product. This means that $\boldsymbol{\mu}$ produced by Theorem 1.8 is a *mixture* of T product distributions. Correlation arises by playing multiple iterations of the game. As a special case, Theorem 1.8 implies that players minimizing swap regret converge—in terms of the average correlated distribution of play—to correlated equilibria, whereas external regret is associated with *coarse* correlated equilibria.

Proof of Theorem 1.8. For any player $i \in [n]$, we have

$$\begin{aligned}
\Phi\mathrm{Reg}^{(T)} &= \max_{\phi_i\in\Phi_i} \left\{ \sum_{t=1}^T \langle \phi(\boldsymbol{x}_i^{(t)}), \boldsymbol{u}_i^{(t)} \rangle \right\} - \sum_{t=1}^T \langle \boldsymbol{x}_i^{(t)}, \boldsymbol{u}_i^{(t)} \rangle \\
&= \max_{\phi_i\in\Phi_i} \left\{ \sum_{t=1}^T u_i(\phi_i(\boldsymbol{x}_i^{(t)}), \boldsymbol{x}_{-i}^{(t)}) \right\} - \sum_{t=1}^T u_i(\boldsymbol{x}_1^{(t)}, \dots, \boldsymbol{x}_n^{(t)}), \qquad (3)
\end{aligned}$$

by multilinearity. Let $\boldsymbol{\mu} = \frac{1}{T}\sum_{t=1}^T \bigotimes_{i=1}^n \boldsymbol{x}_i^{(t)}$. Continuing from (3),

$$\frac{1}{T}\Phi\mathrm{Reg}^{(T)} = \max_{\phi_i\in\Phi_i} \mathbb{E}_{(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)\sim\boldsymbol{\mu}} u_i(\phi_i(\boldsymbol{x}_i), \boldsymbol{x}_{-i}) - \mathbb{E}_{(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)\sim\boldsymbol{\mu}} u_i(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n).$$

□

2 A framework for minimizing Φ-regret

Having connected Φ-regret with correlated equilibrium concepts, we now introduce the elegant framework of Gordon et al. [2008] to minimize Φ-regret in the online learning setting; by virtue of Theorem 1.8, one can then compute a Φ-equilibrium by having each player employ the algorithm of Gordon et al. [2008].

Reducing Φ-regret to external regret The upshot of the construction of Gordon et al. [2008] is that one can reduce Φ-regret to external regret, albeit with some important caveats. We recall that the basic goal is to produce a sequence of strategies that minimizes Φ-regret per (2); Φ is assumed to be convex and compact. The framework of Gordon et al. [2008] provides a general template for doing that. It asks for two basic subroutines.

1. A *fixed-point oracle*: for any deviation $\phi \in \Phi$, it outputs a fixed point $\mathcal{X} \ni \boldsymbol{x} = \phi(\boldsymbol{x})$.
2. An online algorithm $\mathfrak{R}_\Phi$ minimizing *external regret* with respect to *the set* Φ.

With regard to Item 1, we will for now assume that Φ consists of continuous functions mapping $\mathcal{X}$ to $\mathcal{X}$, so that the *existence* of a fixed point is guaranteed by Brouwer's fixed-point theorem. But whether such a fixed point can be computed efficiently is a different story. In fact, computing approximate fixed points of general functions is known to be equivalent to finding Nash equilibria [Daskalakis et al., 2008], which defeats the purpose. For the time being, we can assume that Φ is structured enough so that it admits an efficient fixed-point oracle; for example, this is so when Φ contains only linear deviations. A final noteworthy point about Item 1 is that it will be enough if one has instead an *approximate* fixed-point oracle, in that $\|\boldsymbol{x} - \phi(\boldsymbol{x})\| \leq \epsilon$.

Assuming access to a fixed-point oracle, the reduction of Gordon et al. [2008] reduces Φ-regret to external regret, but with an important catch: the algorithm minimizing external regret needs to operate over the set of deviations Φ (Item 2). This is a significantly more complex set than the one we started with, $\mathcal{X}$, and will be the key to establishing efficient Φ-regret minimizers.

In any event, assuming access to the oracles posited in Item 1 and Item 2, the algorithm of Gordon et al. [2008] (Algorithm 1) produces a Φ-regret minimizer $\mathfrak{R}$ as follows.

- In every time $t \in [T]$, it obtains the next strategy $\phi^{(t)}$ of $\mathfrak{R}_\Phi$. $\mathfrak{R}$ then produces as the next strategy $\boldsymbol{x}^{(t)} \in \mathcal{X}$ any fixed point of $\phi^{(t)}$ through the fixed-point oracle.
- Next, upon observing $\boldsymbol{u}^{(t)}$, $\mathfrak{R}$ feeds to $\mathfrak{R}_\Phi$ the utility function $u_\Phi^{(t)} : \phi \mapsto \langle \phi(\boldsymbol{x}^{(t)}), \boldsymbol{u}^{(t)} \rangle$.

Theorem 2.1 (Gordon et al., 2008)**.** *If* $\mathrm{Reg}^{(T)}$ *is the external regret of* $\mathfrak{R}_\Phi$ *and* $\Phi\mathrm{Reg}^{(T)}$ *is the* Φ*-regret of* $\mathfrak{R}$*, then* $\mathrm{Reg}^{(T)} = \Phi\mathrm{Reg}^{(T)}$.

Proof. We have

$$\begin{aligned}\Phi\mathrm{Reg}^{(T)} &= \max_{\phi \in \Phi} \left\{ \sum_{t=1}^{T} \langle \phi(\boldsymbol{x}^{(t)}), \boldsymbol{u}^{(t)} \rangle \right\} - \sum_{t=1}^{T} \langle \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \rangle \\ &= \max_{\phi \in \Phi} \left\{ \sum_{t=1}^{T} \langle \phi(\boldsymbol{x}^{(t)}), \boldsymbol{u}^{(t)} \rangle \right\} - \sum_{t=1}^{T} \langle \phi^{(t)}(\boldsymbol{x}^{(t)}), \boldsymbol{u}^{(t)} \rangle \end{aligned} \tag{4}$$

since $\boldsymbol{x}^{(t)} = \phi^{(t)}(\boldsymbol{x}^{(t)})$. Continuing from (4),

$$\Phi\mathrm{Reg}^{(T)} = \max_{\phi \in \Phi} \left\{ \sum_{t=1}^{T} u_{\Phi}^{(t)}(\phi) \right\} - \sum_{t=1}^{T} u_{\Phi}^{(t)}(\phi^{(t)}) = \mathrm{Reg}^{(T)}.$$

□

Algorithm 1: The template of Gordon et al. [2008] for minimizing Φ-regret.

Input: An external regret minimizer $\mathfrak{R}_\Phi$ for the set Φ
NextStrategy():
    Set $\phi^{(t)} \coloneqq \mathfrak{R}_\Phi$.NextStrategy();
    return $\mathcal{X} \ni \boldsymbol{x}^{(t)} = \phi^{(t)}(\boldsymbol{x}^{(t)})$;
ObserveUtility($\boldsymbol{u}^{(t)}$):
    Set $u_\Phi^{(t)} : \phi \mapsto \langle \phi(\boldsymbol{x}^{(t)}), \boldsymbol{u}^{(t)} \rangle$;
    $\mathfrak{R}_\Phi$.ObserveUtility($u_\Phi^{(t)}$);

2.1 The algorithm of Blum and Mansour

We will now see how to make use of the previous framework so as to minimize swap regret in a normal-form game setting. The resulting algorithm was first developed by Blum and Mansour [2007] (we also refer to a closely related algorithm by Stoltz and Lugosi, 2005). The key to applying the framework of Gordon et al. [2008] is to understand the structure of the set of deviations Φ. In the special case of normal-form games, it is enough to consider only linear functions mapping $\Delta(\mathcal{A})$ to $\Delta(\mathcal{A})$, for which there is a simple combinatorial characterization in terms of *(column)-stochastic* matrices.

Lemma 2.2. *Any linear function $\phi : \Delta(\mathcal{A}) \to \Delta(\mathcal{A})$ can be equivalently expressed as $\boldsymbol{x} \mapsto \mathbf{M}\boldsymbol{x}$ for some stochastic matrix $\mathbf{M}$.*

Indeed, since ϕ is linear it can be expressed as $\boldsymbol{x} \mapsto \mathbf{M}\boldsymbol{x}$ for some matrix $\mathbf{M}$. Now, every column of $\mathbf{M}$ is equal to the output of ϕ for the probability distribution that places all the probability in the corresponding action profile. But, given that ϕ maps $\Delta(\mathcal{A})$ to $\Delta(\mathcal{A})$, this implies that every column of $\mathbf{M}$ is a probability distribution.

Armed with the characterization of Lemma 2.2, let's now see how to implement the two oracles required in the framework of Gordon et al. [2008]. First, in relation to Item 1, a stochastic matrix induces a Markov chain over $\mathcal{A}$. Any *stationary distribution* of that Markov chain is a fixed point, which can be computed efficiently since it boils down to solving a linear system. (More broadly, when the underlying set $\mathcal{X}$ is a polytope and Φ comprises linear deviations, computing a fixed point amounts to solving a linear program, which can be done in polynomial time.)

Moving on to Item 2, we will next show how to minimize regret with respect to the set of stochastic matrices. By definition, the set of stochastic matrices is a product of simplices—one probability distribution for each column:

$$\{[(\boldsymbol{x}_a)_{a \in \mathcal{A}}] : \boldsymbol{x}_a \in \Delta(\mathcal{A}) \quad \forall a \in \mathcal{A}\},$$

where, for $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{\mathcal{A}}$, $[(\boldsymbol{x}, \boldsymbol{x}')]$ denotes the matrix with columns $\boldsymbol{x}$ and $\boldsymbol{x}'$. But, as we saw last time, minimizing (external) regret over such a set can be accomplished using the regret circuit for the Cartesian product: simply have an independent regret minimizer for each column.

Lemma 2.3. *There is an efficient no-regret algorithm for minimizing external regret over the set of stochastic matrices.*

The overall construction is given in Algorithm 2. It consists of $|\mathcal{A}|$ separate regret minimizers, $(\mathfrak{R}_a)_{a \in \mathcal{A}}$, each of which operates over $\Delta(\mathcal{A})$. To obtain the next strategy, we create the stochastic matrix $\mathbf{M}^{(t)}$ in which each column is given by the strategy of the corresponding regret minimizer (Line 5), and then output any fixed point of $\mathbf{M}^{(t)}$ (Line 6). To explain the second part of the algorithm, let's first note that the utility observed by $\mathfrak{R}_\Phi$, per the construction in Theorem 2.1, can be cast as $u_\Phi(\phi) = \langle \phi(\boldsymbol{x}^{(t)}), \boldsymbol{u}^{(t)} \rangle = \langle \mathbf{M}\boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \rangle = \langle \mathbf{M}, \boldsymbol{u}^{(t)} \otimes \boldsymbol{x}^{(t)} \rangle$, where we used that $\phi(\boldsymbol{x}^{(t)}) = \mathbf{M}\boldsymbol{x}^{(t)}$. In other words, $\mathfrak{R}_\Phi$ observes the utility vector $\boldsymbol{u}^{(t)} \otimes \boldsymbol{x}^{(t)}$. Per the regret circuit of the Cartesian product, one should forward to each $\mathfrak{R}_a$ its corresponding component, which is $\boldsymbol{x}^{(t)}[a]\boldsymbol{u}^{(t)}$ (Line 9). If we instantiate each $\mathfrak{R}_a$ with MWU and invoke Theorem 2.1, we arrive at the following result.

Theorem 2.4. *There is an online algorithm whose swap regret is bounded by* $O(\sqrt{T|\mathcal{A}| \log |\mathcal{A}|})$.

The naive argument here would give $O(|\mathcal{A}|\sqrt{T \log |\mathcal{A}|})$ since each MWU algorithm incurs an external regret bounded by $O(\sqrt{T \log |\mathcal{A}|})$, but one can make use of the structure of the observed utilities to obtain the improved bound claimed in Theorem 2.4. The basic reason is that, for any $t \in [T]$,

$$\sum_{a \in \mathcal{A}} \|\boldsymbol{u}_a^{(t)}\|_\infty^2 = \|\boldsymbol{u}^{(t)}\|_\infty^2 \sum_{a \in \mathcal{A}} (\boldsymbol{x}^{(t)}[a])^2 \leq \|\boldsymbol{u}^{(t)}\|_\infty^2.$$

So, using the regret bound of MWU together with Theorem 2.1,

$$\Phi\mathrm{Reg}^{(T)} \leq \frac{|\mathcal{A}| \log |\mathcal{A}|}{\eta} + \eta \sum_{t=1}^{T} \|\boldsymbol{u}^{(t)}\|_\infty^2 \leq \frac{|\mathcal{A}| \log |\mathcal{A}|}{\eta} + \eta T.$$

Optimizing the learning rate η gives the claim.

Algorithm 2: Blum-Mansour algorithm for minimizing swap regret

1 **Input:** A regret minimizer $\mathfrak{R}_a$ for each action $a \in \mathcal{A}$
2 NextStrategy():
3 **for** *each action* $a \in \mathcal{A}$ **do**
4 $\Delta(\mathcal{A}) \ni \boldsymbol{x}_a^{(t)} \coloneqq \mathfrak{R}_a$.NextStrategy();
5 Set $\mathbf{M}^{(t)} \coloneqq [(\boldsymbol{x}_a^{(t)})_{a \in \mathcal{A}}]$;
6 **return** $\Delta(\mathcal{A}) \ni \boldsymbol{x}^{(t)} = \mathbf{M}^{(t)}\boldsymbol{x}^{(t)}$;
7 ObserveUtility($\boldsymbol{u}^{(t)} \in \mathbb{R}^{\mathcal{A}}$):
8 **for** *each action* $a \in \mathcal{A}$ **do**
9 Set $\boldsymbol{u}_a^{(t)} \coloneqq \boldsymbol{x}^{(t)}[a]\boldsymbol{u}^{(t)}$;
10 $\mathfrak{R}_a$.ObserveUtility($\boldsymbol{u}_a^{(t)}$);

References

John Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36:48–49, 1950.

Constantinos Daskalakis, Paul Goldberg, and Christos Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 2008.

Itzhak Gilboa and Eitan Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1:80–93, 1989.

H. Moulin and J.-P. Vial. Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, 7(3-4):201–221, 1978.

Robert Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67–96, 1974.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Geoffrey J Gordon, Amy Greenwald, and Casey Marks. No-regret learning in convex games. In *International Conference on Machine Learning*, 2008.

Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8:1307–1324, 2007.

Gilles Stoltz and Gábor Lugosi. Internal regret in on-line portfolio selection. *Machine Learning*, 59(1-2):125–159, 2005.