Lecture 7
Algorithms for Minimizing Φ-Regret: Nonlinear Deviations and Ellipsoid

Ioannis Anagnostides

Our last lecture introduced the general framework of Gordon et al. [2008] for minimizing Φ-regret, which is predicated on two basic components: i) a fixed-point oracle and ii) an external regret minimizer for the set Φ. But is it always possible to efficiently implement those oracles? If not, are they really necessary for minimizing Φ-regret?

To answer these questions, we will first discuss, in Section 1, a result of Hazan and Kale [2007] which shows that, in a *certain* sense, computing fixed points is *necessary*, proving a partial converse to the result of Gordon et al. [2008]. This is quite unfortunate, as computing fixed points of general functions is a computationally hard problem [Daskalakis et al., 2008]. Even so, we will see how one can bypass the lower bound of Hazan and Kale [2007] by allowing the learner to output *mixed strategies*. We will then have a small interlude to introduce the famous *ellipsoid against hope* algorithm of Papadimitriou and Roughgarden [2008] (Section 2), highlighting several similarities between that algorithm and the one by Gordon et al. [2008]. We will then turn our attention to the second component required in the template of Gordon et al. [2008]. We will see how one can minimize Φ-regret even when it's impossible to efficiently minimize external regret over Φ (Section 3). In other words, both of the oracles we discussed last time can be relaxed.

1 Expected fixed points

The first basic oracle posited in the framework of Gordon et al. [2008] is an algorithm that outputs, for any deviation $\phi \in \Phi$, a *fixed point* of ϕ.

Necessity of fixed points An immediate conceptual question is whether such an oracle is really necessary. Hazan and Kale [2007] have shown that, in a *certain* sense, the answer is yes, as formalized below.

Proposition 1.1 (Hazan and Kale, 2007). *If there a polynomial-time algorithm that outputs strategies in $\mathcal{X}$ with strongly sublinear Φ-regret, there is an* FPTAS *for computing a fixed point of any $\phi \in \Phi$.*

We recall that an online algorithm is said to have strongly sublinear (Φ-)regret if it grows as $T^{1-\alpha}$ for some constant $\alpha \in (0, 1]$. A fully polynomial-time approximation scheme (FPTAS) is an algorithm whose running time is polynomial in the dimension of $\mathcal{X}$ and $1/\epsilon$.

Proof of Proposition 1.1. Let $\epsilon > 0$ be the desired precision and $\phi \in \Phi$ any function whose fixed point is to be computed. At any time $t \in [T]$, let $\boldsymbol{x}^{(t)} \in \mathcal{X}$ be the output of the algorithm.

If $\|\phi(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}\|_2 \le \epsilon$ we can terminate. Otherwise, we forward to the algorithm the utility function

$$u^{(t)} : \boldsymbol{x} \mapsto \frac{1}{\|\phi(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}\|_2} \langle \phi(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}, \boldsymbol{x} - \boldsymbol{x}^{(t)} \rangle.$$

Suppose that the algorithm failed to terminate with an ϵ-fixed point after the T rounds. Then its Φ-regret reads

$$\Phi\mathrm{Reg}^{(T)} \ge \sum_{t=1}^{T} u^{(t)}(\phi(\boldsymbol{x}^{(t)})) - \sum_{t=1}^{T} u^{(t)}(\boldsymbol{x}^{(t)}) = \sum_{t=1}^{T} \|\phi(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}\|_2 \ge \epsilon T.$$

When T is large enough, this contradicts the assumption that the algorithm is minimizing Φ-regret. □

Proposition 1.1 should be seen as an obstacle to minimizing Φ-regret, given that computing fixed points of general, nonlinear functions is intractable—in fact, equivalent to computing Nash equilibria [Daskalakis et al., 2008]. But should we even care about Φ-regret beyond the setting where Φ contains linear functions?

The importance of nonlinear deviations While in normal-form games restricting to linear deviations suffices to encompass all possible deviations from pure strategies to pure strategies, as was covered in the last lecture, it will not be enough beyond that setting.

Example 1.2 (XOR). Let $\mathcal{X} = [0,1]^2$ be the strategy set of a learner in a tree-form decision problem. Consider the strategy deviation

$$\begin{pmatrix}0\\1\end{pmatrix} \mapsto \begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix} \mapsto \begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix} \mapsto \begin{pmatrix}1\\1\end{pmatrix}, \text{ and } \begin{pmatrix}0\\0\end{pmatrix} \mapsto \begin{pmatrix}*\\*\end{pmatrix}.$$

We claim there is *no* linear deviation $[0,1]^2 \to [0,1]^2$ that captures this deviation. For the sake of contradiction, let $\boldsymbol{x} \mapsto \mathbf{M}\boldsymbol{x}$ be a purported such linear function. We have

$$\mathbf{M}\left(\frac{1}{2}\begin{pmatrix}0\\1\end{pmatrix} + \frac{1}{2}\begin{pmatrix}1\\0\end{pmatrix}\right) = \frac{1}{2}\mathbf{M}\begin{pmatrix}0\\1\end{pmatrix} + \frac{1}{2}\mathbf{M}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}.$$

But, at the same time,

$$\mathbf{M}\left(\frac{1}{2}\begin{pmatrix}0\\1\end{pmatrix} + \frac{1}{2}\begin{pmatrix}1\\0\end{pmatrix}\right) = \frac{1}{2}\mathbf{M}\begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1\\1\end{pmatrix},$$

a contradiction.

The role of mixed strategies Fortunately, there is a seemingly innocuous precondition in the statement of Proposition 1.1 that makes all the difference: the learner must output strategies in $\mathcal{X}$. It turns out that one can sidestep Proposition 1.1 when the learner is able to output *mixed strategies*—points in $\Delta(\mathcal{X})$.

Now, to connect this discrepancy with an earlier lecture, Kuhn's theorem would seem to imply that mixed strategies are no more powerful than sequence-form (or behavioral) strategies—we are implicitly assuming perfect recall. But it is precisely the nonlinear nature of the deviations that renders that equivalence invalid in this setting.

Expected fixed points This discrepancy was recently observed by Zhang et al. [2024] who introduced the notion of an *expected fixed point.*

Definition 1.3 (Zhang et al., 2024)**.** For a function $\phi : \mathcal{X} \to \mathcal{X}$, a distribution $\boldsymbol{\mu} \in \Delta(\mathcal{X})$ is an *ϵ-expected fixed point* of ϕ if

$$\left\| \mathbb{E}_{\boldsymbol{x} \sim \boldsymbol{\mu}} [\phi(\boldsymbol{x}) - \boldsymbol{x}] \right\| \leq \epsilon.$$

There are two key points about expected fixed points. First, unlike actual fixed points, an approximate expected fixed point *can be computed efficiently.* In particular, the following simple algorithm suffices. Start from any point $\boldsymbol{x}_0 \in \mathcal{X}$ and repeated apply the mapping $\boldsymbol{x}_k \leftarrow \phi(\boldsymbol{x}_{k-1})$ for $k = 1, \dots, K$. The key observation is that the resulting summation telescopes:

$$\sum_{k=1}^{K} (\phi(\boldsymbol{x}_k) - \boldsymbol{x}_k) = \phi(\boldsymbol{x}_K) - \boldsymbol{x}_1 + \sum_{k=1}^{K-1} (\phi(\boldsymbol{x}_k) - \boldsymbol{x}_{k+1}) = \phi(\boldsymbol{x}_K) - \boldsymbol{x}_1.$$

So, if we select the uniform distribution $\boldsymbol{\mu}$ over $\boldsymbol{x}_1, \dots, \boldsymbol{x}_K$, we have $\mathbb{E}_{\boldsymbol{x} \sim \mu}[\phi(\boldsymbol{x}) - \boldsymbol{x}] = \frac{1}{K}(\phi(\boldsymbol{x}_K) - \boldsymbol{x}_1)$. In other words, by taking the length of the sequence K large enough, we can compute an ϵ-expected fixed point after $O(1/\epsilon)$ repetitions. It turns out that there is also an exponentially faster algorithm for computing expected fixed points whose complexity scales polynomially in $\log(1/\epsilon)$ [Zhang et al., 2025], but we will not cover it here. (The upshot is that it's based on an application of the EAH algorithm, covered in Section 2; one can try to prove this as an exercise.)

Theorem 1.4 (Zhang et al., 2025)**.** *There is a polynomial-time algorithm for computing an expected fixed point of any function $\phi : \mathcal{X} \to \mathcal{X}$.*

The second key point about expected fixed points is that they can be used *in lieu* of fixed points in the construction of Gordon et al. [2008], otherwise Theorem 1.4 would be moot. Let's now spell out the missing details. When the learner is instead outputting a distribution $\boldsymbol{\mu}^{(t)} \in \Delta(\mathcal{X})$ in every round, we define its Φ-regret as

$$\Phi\mathsf{Reg}^{(T)} \coloneqq \max_{\phi \in \Phi} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x}^{(t)} \sim \boldsymbol{\mu}^{(t)}} \langle \phi(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \rangle.$$

As in the previous lecture, we still assume that we have an algorithm $\mathfrak{R}_\Phi$ that minimizes external regret over the set of deviations Φ. The key difference is that, instead of returning an actual fixed point of $\phi^{(t)}$, we return an *expected* fixed point (Line 4).

Theorem 1.5. *If* $\mathsf{Reg}^{(T)}$ *is the external regret of* $\mathfrak{R}_\Phi$*, then the* Φ*-regret can be bounded as*

$$\Phi\mathsf{Reg}^{(T)} \leq \mathsf{Reg}^{(T)} + \epsilon B T,$$

where $\epsilon > 0$ is a bound on the expected fixed-point gap (per Definition 1.3) and $\|\boldsymbol{u}^{(t)}\|_* \leq B$.

Algorithm 1: A refinement of Gordon et al. [2008] using expected fixed points.

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

Proof. We have

$$\begin{aligned}
\Phi\mathrm{Reg}^{(T)} &= \max_{\phi \in \Phi} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x}^{(t)} \sim \boldsymbol{\mu}^{(t)}} \langle \phi(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \rangle \\
&= \max_{\phi \in \Phi} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x}^{(t)} \sim \boldsymbol{\mu}^{(t)}} \langle \phi(\boldsymbol{x}^{(t)}) - \phi^{(t)}(\boldsymbol{x}^{(t)}), \boldsymbol{u}^{(t)} \rangle + \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x}^{(t)} \sim \boldsymbol{\mu}^{(t)}} \langle \phi^{(t)}(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \rangle \\
&\le \max_{\phi \in \Phi} \sum_{t=1}^{T} (u_\Phi^{(t)}(\phi) - u_\Phi^{(t)}(\phi^{(t)})) + \sum_{t=1}^{T} \left\| \mathbb{E}_{\boldsymbol{x}^{(t)} \sim \boldsymbol{\mu}^{(t)}} (\phi^{(t)}(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}) \right\| \|\boldsymbol{u}^{(t)}\|_* \\
&\le \mathrm{Reg}^{(T)} + \epsilon B T.
\end{aligned}$$

□

The main takeaway is that minimizing Φ-regret *only* requires a no-external-regret algorithm with respect to the set Φ. Section 3 explores the complexity of that latter problem, and presents a further, crucial relaxation in the framework of Gordon et al. [2008]. But first, we interpose a celebrated algorithm for computing correlated equilibria; as we shall see, it's intimately connected with the framework of Gordon et al. [2008].

2 The ellipsoid against hope algorithm: solving the correlated equilibrium LP in multi-player games

The techniques we have covered based on regret minimization lead to a running time complexity growing polynomially in $1/\epsilon$, where $\epsilon > 0$ is the approximation quality of the equilibrium—be it a Nash equilibrium in zero-sum games or a (coarse) correlated equilibrium in general-sum games. Specifically, if we employ algorithms with (Φ-)regret bounded by $\sqrt{T}$ as a function of the time horizon, we need $1/\epsilon^2$ iterations to reach an equilibrium. Can we do better in the regime where $\epsilon \ll 1$? The other main approach we have introduced for equilibrium computation is based on linear programming, which has the advantage of finding an *exact* equilibrium. But, as we saw in the last lecture, the LP describing (C)CEs has a number of variables that scales *exponentially* with the number of players.

We will now introduce *ellipsoid against hope (*EAH*)*, a polynomial-time algorithm for computing (C)CEs even in multi-player games. It was first introduced by Papadimitriou and Roughgarden [2008]; in what follows, we mostly follow the generalized version of the algorithm due to Farina and Pipis [2024]. To be clear, this algorithm works in the centralized model, and it's not compatible with the framework of online learning, although similarities do exist between the two approaches as we shall see.

Our goal is to compute an ϵ-Φ-*equilibrium* of a multilinear game; that is, a correlated distribution $\boldsymbol{\mu} \in \Delta(\mathcal{X}_1 \times \cdots \times \mathcal{X}_n)$ such that for any player $i \in [n]$ and deviation function $\Phi_i \ni \phi_i : \mathcal{X}_i \to \mathcal{X}_i$,

$$\mathbb{E}_{(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)\sim\mu} u_i(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n) \geq \mathbb{E}_{(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)\sim\mu} u_i(\phi_i(\boldsymbol{x}_i), \boldsymbol{x}_{-i}) - \epsilon. \tag{1}$$

The ellipsoid against hope framework From a more abstract point of view, EAH deals with optimization problems of the form[1]

$$\text{find } \boldsymbol{\mu} \in \Delta(\mathcal{X}) \text{ such that } \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\mu}} \langle \boldsymbol{y}, G(\boldsymbol{x}) \rangle \geq 0 \quad \forall \boldsymbol{y} \in \mathcal{Y}, \tag{2}$$

where $\mathcal{X} \subseteq \mathbb{R}^d, \mathcal{Y} \subseteq \mathbb{R}^k$, and $G : \mathcal{X} \to \mathbb{R}^k$ is a function that can be evaluated efficiently. The crux in the optimization problem (2) lies in the fact that $\boldsymbol{\mu}$ resides in a high-dimensional space; a canonical case to think about is when $\mathcal{X} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ in a normal-form game, so that even describing a distribution $\boldsymbol{\mu}$ could require specifying $\prod_{i=1}^n |\mathcal{A}_i| - 1$ coordinates. We assume that $\mathcal{Y}$, which corresponds to the set of deviations, admits a *separation oracle*; under mild geometric assumptions, it's equivalent to posit merely a *membership oracle*, which returns whether a point $\boldsymbol{y} \in \mathbb{R}^k$ belongs to $\mathcal{Y}$ or not. A special case is when $\mathcal{Y}$ is a polytope described with a polynomial number of constraints, as is the case for swap regret in normal-form games—each player's set of deviations amounts to the set of stochastic matrices. The key assumption in the EAH framework [Farina and Pipis, 2024] is the admission of a *good-enough-response (GER)* oracle, which, *given* any $\boldsymbol{y} \in \mathcal{Y}$, returns a point $\boldsymbol{x} \in \mathcal{X}$ such that $\langle \boldsymbol{y}, G(\boldsymbol{x}) \rangle \geq 0$.

The EAH algorithm enables solving (2) with just a separation oracle for $\mathcal{Y}$ and a GER oracle. The basic idea is to consider an ϵ-approximate version of the dual of (2),

$$\text{find } \boldsymbol{y} \in \mathcal{Y} \text{ such that } \langle \boldsymbol{y}, G(\boldsymbol{x}) \rangle \leq -\epsilon \quad \forall \boldsymbol{x} \in \mathcal{X}. \tag{3}$$

On account of the fact that a GER oracle exists, (3) is guaranteed to be infeasible. Even so, EAH proceeds by executing the ellipsoid algorithm on that infeasible program—this is where the apt name "ellipsoid against hope" comes from. In every step t of the algorithm, we have a candidate $\boldsymbol{y}^{(t)} \in \mathcal{Y}$, and use the GER oracle to produce a point $\boldsymbol{x}^{(t)} \in \mathcal{X}$ that refutes $\boldsymbol{y}^{(t)}$; in fact, an entire halfspace in $\mathbb{R}^k$. This goes on until the volume of the ellipsoid has shrank to a small enough amount. It then follows that

$$\forall \boldsymbol{y} \in \mathcal{Y} \exists t \in [T] \text{ such that } \langle \boldsymbol{y}, G(\boldsymbol{x}^{(t)}) \rangle > -\epsilon.$$

[1] We caution that $\mathcal{X}$ here corresponds to $\mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ in the context of (1), even though we usually take $\mathcal{X}$ to be the strategy set of a single player in the context of regret minimization.

Thus,

$$\min_{\boldsymbol{y}\in\mathcal{Y}} \max_{\boldsymbol{\mu}\in\Delta([T])} \sum_{t=1}^{T} \boldsymbol{\mu}^{(t)} \langle \boldsymbol{y}, G(\boldsymbol{x}^{(t)}) \rangle > -\epsilon.$$

By the minimax theorem, we conclude that

$$\max_{\boldsymbol{\mu}\in\Delta([T])} \min_{\boldsymbol{y}\in\mathcal{Y}} \sum_{t=1}^{T} \boldsymbol{\mu}^{(t)} \langle \boldsymbol{y}, G(\boldsymbol{x}^{(t)}) \rangle > -\epsilon. \tag{4}$$

In other words, there is a convex combination of $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(T)}$ that refutes any possible $\boldsymbol{y} \in \mathcal{Y}$. That convex combination, which is *certificate of dual infeasibility*, will thus be an approximate solution to (2). The key point is that the resulting zero-sum game in (4) is much smaller than the one we started with, and can be solved with standard LP techniques.

Theorem 2.1 (Farina and Pipis, 2024). *Assuming the existence of a separation oracle for $\mathcal{Y}$ and a GER oracle,* EAH *runs in time* $\mathsf{poly}(d, k, \log(1/\epsilon))$ *and returns an ϵ-approximate solution to* (2).

Let's now see to apply this algorithm to solve (1). We assume that each Φ_i contains linear functions of the form $\mathcal{X}_i \ni \boldsymbol{x}_i \mapsto \mathbf{M}_i \boldsymbol{x}_i \in \mathcal{X}_i \subseteq \mathbb{R}^{m_i}$, where the set of valid matrices $\mathbf{M}_i$ is a polytope $\mathcal{Y}_i$ with a polynomial number of variables and constraints. (For example, in the example covered in the last lecture pertaining to swap regret in normal-form games, the set of column stochastic matrices is a Cartesian product of probability simplices.) We assume further that Φ_i contains the identity matrix. Then (1) can be expressed as

$$\mathbb{E}_{(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)\sim\mu} \langle \mathbf{I} - \mathbf{M}_i, \boldsymbol{u}_i(\boldsymbol{x}_{-i}) \otimes \boldsymbol{x}_i \rangle \geq -\epsilon \quad \forall i \in [n], \mathbf{M}_i \in \mathcal{Y}_i, \tag{5}$$

where $\mathbf{I}$ denotes the identity matrix, since

$$\mathbb{E}_{(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)\sim\mu} \langle \boldsymbol{x}_i - \mathbf{M}_i \boldsymbol{x}_i, \boldsymbol{u}_i(\boldsymbol{x}_{-i}) \rangle = \mathbb{E}_{(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)\sim\mu} \langle \mathbf{I} - \mathbf{M}_i, \boldsymbol{u}_i(\boldsymbol{x}_{-i}) \otimes \boldsymbol{x}_i \rangle.$$

This indeed adheres to (2). What's left is to prove that (5) admits a GER oracle. To do so, let's consider any $i \in [n]$ and deviation $\mathbf{M}_i \in \mathcal{Y}_i$. Taking $\boldsymbol{x}_i \in \mathcal{X}_i$ to be a *fixed point* of $\mathbf{M}_i$, we have

$$\mathbb{E}_{(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)\sim\mu} \langle \boldsymbol{x}_i - \mathbf{M}_i \boldsymbol{x}_i, \boldsymbol{u}_i(\boldsymbol{x}_{-i}) \rangle = 0,$$

as desired.

Theorem 2.2 (Farina and Pipis, 2024). *If for each player $i \in [n]$ in a multilinear game the set of deviations Φ_i admits a separation oracle, there is an algorithm polynomial in* $\log(1/\epsilon)$*, n, and $m = \max_{1\leq i\leq n} m_i$ that outputs an ϵ-Φ-equilibrium.*

Overall, we find that EAH and the framework of Φ-regret have several conceptual similarities. Namely, they both operate over the set of deviations, returning in each round a fixed point of the corresponding deviation. Further, the correlated distribution they output is given as a mixture of product distributions. This ensures that the correlated distribution admits an efficient, compact representation. The final similarity is that they both require a separation oracle for the set of deviations, which brings us to the final topic of this lecture.

3 Nonlinear deviations and semi-separation oracle

The main requirement in both EAH and the framework of Gordon et al. [2008] is a separation oracle for the set of deviations. Indeed, focusing on minimizing external regret, a separation oracle is enough to efficiently implement an algorithm such as FTRL; conversely, any regret minimizer should be able to perform separation; otherwise, there would be a fixed utility exerting high regret.

However, even when Φ contains only linear functions mapping $\mathcal{X} \to \mathcal{X}$, it can be NP-hard to implement a membership oracle, as shown recently by Daskalakis et al. [2025]. Does this mean that minimizing Φ-regret or applying the EAH framework is intractable? Not so. While so far we have insisted on having *both* a separation oracle for the set of deviations *and* an expected fixed-point oracle (that is, a good-enough-response oracle), it turns out that it's enough to implement their *disjunction*; this was coined *semi-separation* oracle by Daskalakis et al. [2025] (Definition 3.2).

We will introduce this notion in the general setting where the set Φ comprises nonlinear functions expressed in the following form.

Definition 3.1 (Zhang et al., 2025)**.** Given a map $\psi : \mathcal{X} \to \mathbb{R}^{k'}$, the set of deviations Φ^ψ is the set of all maps $\phi : \mathcal{X} \to \mathcal{X}$ that can be expressed as the matrix-vector product $\mathbf{K}(\phi)\psi(\boldsymbol{x}) + \boldsymbol{c}(\phi)$ for some matrix $\mathbf{K} \in \mathbb{R}^{d \times k'}$ and vector $\boldsymbol{c} \in \mathbb{R}^d$. We denote by $k = d \times k' + d$ the dimension of $(\mathbf{K}, \boldsymbol{c})$.

One can think of ψ as a *feature mapping*; a natural example captured by Definition 3.1 is *low-degree polynomials*, where ψ contains all $\leq \ell$-wise products of entries in $\boldsymbol{x}$. Minimizing regret with respect to the set describing all valid $(\mathbf{K}, \boldsymbol{c})$ is daunting, but it turns out that this is not necessary. One can instead solve the following simpler problem; the reason why this is sufficient is not covered in these notes.

Definition 3.2 (Semi-separation)**.** In the *semi-separation* problem we are given a function $\phi : \mathcal{X} \to \mathbb{R}^d$ and we have to compute

- *either* an ϵ-expected fixed point $\boldsymbol{\mu} \in \Delta(\mathcal{X})$ of ϕ,
- *or* a point $\boldsymbol{x} \in \mathcal{X}$ such that $\phi(\boldsymbol{x}) \notin \mathcal{X}$.

Implementing the semi-separation oracle is actually not hard once we have an algorithm for computing expected fixed points (Theorem 1.4). One can basically run the same algorithm: if during the execution of the algorithm we identify a point such that $\phi(\boldsymbol{x}) \notin \mathcal{X}$, we can terminate as we have a certificate that ϕ is not a valid mapping. Otherwise, the algorithm will be able to find an expected fixed point of ϕ, *even though ϕ may not necessarily be a valid transformation.* In other words, the basic idea in the recent framework of Daskalakis et al. [2025] is to make allowance for functions that do not necessarily map $\mathcal{X}$ to $\mathcal{X}$. To conclude, we state the main implication for minimizing Φ^ψ-regret per Definition 3.1.

Theorem 3.3. *There is an algorithm, based on* EAH*, that computes an ϵ-Φ^ψ-equilibrium of any multilinear n-player game and has complexity* $\mathsf{poly}(n, m, k, \log(1/\epsilon))$*. Furthermore, in the online learning setting, there is an efficient algorithm with regret bounded as* $\mathsf{poly}(m, k)\sqrt{T}$ *after T rounds.*

References

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

Elad Hazan and Satyen Kale. Computational equivalence of fixed points and no regret algorithms, and convergence to equilibria. In *Neural Information Processing Systems (NIPS)*, 2007.

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

Christos H. Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 2008.

Brian Hu Zhang, Ioannis Anagnostides, Gabriele Farina, and Tuomas Sandholm. Efficient Φ-regret minimization with low-degree swap deviations in extensive-form games. In *Neural Information Processing Systems*, 2024.

Brian Hu Zhang, Ioannis Anagnostides, Emanuel Tewolde, Ratip Emin Berker, Gabriele Farina, Vincent Conitzer, and Tuomas Sandholm. Learning and computation of Φ-equilibria at the frontier of tractability. In *Economics and Computation*, 2025.

Gabriele Farina and Charilaos Pipis. Polynomial-time computation of exact Φ-equilibria in polyhedral games. In *Neural Information Processing Systems*, 2024.

Constantinos Daskalakis, Gabriele Farina, Maxwell Fishelson, Charilaos Pipis, and Jon Schneider. Efficient learning and computation of linear correlated equilibrium in general convex games. In *Symposium on Theory of Computing (STOC)*, 2025.