

Chapter 2

Beyond Normal Form

Ioannis Anagnostides, Gabriele Farina, and Brian Hu Zhang*

In this section, we will show how to construct Φ -regret minimizers for strategy sets \mathcal{X} that are more complex than the simplex. To do so, we first define the sets of deviations that we work with.

Definition 2.1 ([ZAT+25a]). Given a feature map $q : \mathcal{X} \rightarrow \mathbb{R}^k$, the set of deviations Φ^q is the set of all maps $\varphi_{\mathbf{M}} : \mathcal{X} \rightarrow \mathcal{X}$ that can be expressed as the matrix-vector product $\mathbf{M}(\varphi)q(\mathbf{x})$ for some matrix $\mathbf{M} \in \mathbb{R}^{d \times k}$ and vector $\mathbf{c} \in \mathbb{R}^d$.

We will assume without loss of generality that Φ^q contains the identity map.

Example 2.2. If $q(\mathbf{x}) = (1, \mathbf{x}) \in \mathbb{R}^{d+1}$, then Φ^q is the set of all linear maps $\varphi : \mathcal{X} \rightarrow \mathcal{X}$. More generally, if $q(\mathbf{x}) = (1, \mathbf{x})^{\otimes \ell} \in \mathbb{R}^k$ where $k = \binom{d}{\leq \ell}$ is the set of all monomials in \mathbf{x} up to degree ℓ , then Φ^q is the set of all degree- ℓ polynomials in \mathbf{x} .

In general, we will describe algorithms that have complexity polynomial in the dimension k . Note that, in the special case of degree- ℓ polynomials, these algorithms will therefore run in time polynomial in d^ℓ .

In these settings, we run into two fundamental issues when trying directly to generalize the GGM construction. Recall that the GGM construction requires two ingredients. We will describe why both ingredients are problematic.

1. *Fixed-point oracle:* Finding a fixed point of a continuous function $\varphi : \mathcal{X} \rightarrow \mathcal{X}$ is PPAD-complete, and therefore is unlikely to be possible efficiently. Even more generally, if φ contains *discontinuous* maps, φ may not even admit any approximate fixed points!
2. *External regret minimizer for Φ^q :* External regret minimization implies the ability to (approximately) perform linear optimization over Φ^q . But linear optimization over Φ^q can be hard, in two ways:
 - a. If Φ^q contains nonlinear functions, then that this optimization problem is NP-hard even when \mathcal{X} is the hypercube $[0, 1]^d$ [ZAFS24].

*✉ ianagnos@cs.cmu.edu, {gfarina,zhangbh}@mit.edu. These tutorial notes have not undergone formal peer review. We are grateful for any feedback or reports of typos.

- b. Even if Φ^q is the set of linear endomorphisms, if \mathcal{X} itself is only available via an oracle (e.g., a separation oracle), then optimizing a linear function over Φ^q can in general be hard [DFF+25].

Remark 2.3. When \mathcal{X} is represented by an explicit set of linear constraints, i.e., $\mathcal{X} = \{\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$, the set of linear endomorphisms can also be expressed as an explicit set of linear constraints [ZAT+25b]. This is true, for instance, for *extensive-form games*. Thus, in this particular case, linear regret minimization can be accomplished without all the machinery of this section.

Remark 2.4. For extensive-form games in particular, the set of linear endomorphisms has a clean game-theoretic interpretation [ZFS24]. We will not detail it here, though.

We will show how to circumvent both of the above issues. We start with the fixed point computation.

2.1 Expected fixed points

Given that computing a fixed point is generally hard (when φ is continuous but nonlinear) or even impossible (when φ is discontinuous), we need a different notion of fixed point. Fortunately, by examining the proof of the Gordon–Greenwald–Marks framework, it turns out that a weaker condition suffices.

Definition 2.5 (Expected fixed point [ZAFS24]). Given a function $\varphi : \mathcal{X} \rightarrow \mathcal{X}$, an ϵ -approximate *expected fixed point* of φ is a distribution¹ $\mu \in \Delta(\mathcal{X})$ such that

$$\left\| \mathbb{E}_{\mathbf{x} \sim \mu} [\varphi(\mathbf{x}) - \mathbf{x}] \right\|_2 \leq \epsilon.$$

As we will see shortly, expected fixed points fix both problems with fixed point computation: an (approximate) expected fixed point always exists, and moreover, there is an efficient algorithm for computing one!

Theorem 2.6 (Expected fixed point computation). There is an algorithm with runtime $\text{poly}(d, 1/\epsilon)$ that, given query access to $\varphi : \mathcal{X} \rightarrow \mathcal{X}$ and oracle access to \mathcal{X} , outputs an ϵ -approximate expected fixed point of φ .

The existence of this algorithm proves, as a corollary, that ϵ -approximate expected fixed points always exist for every $\epsilon > 0$. Although we will not need it here, it also turns out to be true that *exact* expected fixed points also always exist (even when φ is discontinuous) [ZAT+25a]. The “either-or” nature of the above theorem will become useful shortly.

¹Throughout this document, to avoid measure-theoretic issues, we only consider finite-support distributions.

Proof. Let $\mathbf{x}^{(1)} \in \mathcal{X}$ be arbitrary, and consider the sequence of points $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(K)}$ where $\mathbf{x}^{(k)} = \varphi(\mathbf{x}^{(k-1)})$ for each $k > 0$. Let μ be the uniform distribution on $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(K)}\}$. Then, by a telescoping sum, we have

$$\mathbb{E}_{\mathbf{x} \sim \mu} [\varphi(\mathbf{x}) - \mathbf{x}] = \frac{1}{K} \sum_{k=1}^K (\varphi(\mathbf{x}^{(k)}) - \mathbf{x}^{(k)}) = \frac{1}{K} \sum_{k=1}^K (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \frac{1}{K} (\mathbf{x}^{(K+1)} - \mathbf{x}^{(1)})$$

where $\mathbf{x}^{(K+1)} := \varphi(\mathbf{x}^{(K)})$ for notational simplicity. But the right-hand side has norm at most $\text{diam}(\mathcal{X})/K$, so setting $K = \text{diam}(\mathcal{X})/\epsilon$ completes the proof. \square

2.2 Semi-separation

Theorem 2.6 also, conveniently, solves the second problem. The key observation, as it turns out, is that it is not actually necessary to be able to efficiently optimize/separate over Φ^q . In fact, it is sufficient to be able to perform the following task, which we call *semi-separation*.

Definition 2.7 (Semi-separation). In the *semi-separation* problem we are given a function $\varphi : \mathcal{X} \rightarrow \mathbb{R}^d$ and we have to compute *either*

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

The proof of Theorem 2.6 also gives a solution to the semi-separation problem: indeed, while constructing the sequence $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(K)}$, if any $\mathbf{x}^{(k)}$ is not in \mathcal{X} , we can immediately output $\mathbf{x}^{(k-1)}$.

It is instructive to contemplate briefly the implications of this result. A point $\mathbf{x} \in \mathcal{X}$ such that $\varphi(\mathbf{x}) \notin \mathcal{X}$ gives a separating direction $\mathbf{a}^\top \varphi(\mathbf{x}) \leq b$ separating $\varphi(\mathbf{x})$ from \mathcal{X} ; the same constraint can be used to separate φ from Φ^q . Determining whether a given function $\varphi : \mathcal{X} \rightarrow \mathbb{R}^d$ actually has image in \mathcal{X} is a hard problem, and indeed a semi-separation algorithm does not achieve this. Semi-separation only guarantees an expected fixed point *or* a separation certificate; in particular, it is possible that semi-separation algorithms output an expected fixed point even when $\varphi \notin \Phi^q$.

2.3 Efficient regret minimization via semi-separation

We now briefly sketch how to use a semi-separation oracle to construct an efficient regret minimizer over any set of functions Φ^q expressible in the form given by Definition 2.1.

At a high level, the algorithm relies on an online algorithm called *Shell gradient descent*, developed by [Daskalakis, Farina, Fishelson, Pipis and Schneider](#) [DFFF+25]. This is just projected gradient descent, but with the twist that the underlying constraint set—the “shell set” $\tilde{\Phi}$ —is *changing from round to round*. The key invariance which guarantees correctness is that $\tilde{\Phi}$ always contains the constraint set of interest—namely, the set of endomorphisms. What’s more, unlike the set of endomorphisms Φ , $\tilde{\Phi}$ always admits efficient oracle access.

Now, for a given function φ , the algorithm of Daskalakis, Farina, Fishelson, Pipis and Schneider [DFF+25] first attempts to find a fixed point of φ in \mathcal{X} through a semi-separation oracle. If it succeeds, we can safely output such a fixed point *even though φ may not necessarily be an endomorphism*. Otherwise, the semi-separation oracle yields a separating direction, thereby providing sufficient information to shrink the shell set $\tilde{\Phi}$. Moreover, shell gradient descent hinges on a projection oracle, which again crucially uses the semi-separation oracle developed above. For the details, we refer the interested readers to Daskalakis, Farina, Fishelson, Pipis and Schneider [DFF+25], as well as Zhang, Anagnostides, Tewolde, Berker, Farina, Conitzer and Sandholm [ZAT+25a] for the extension beyond linear swap regret.

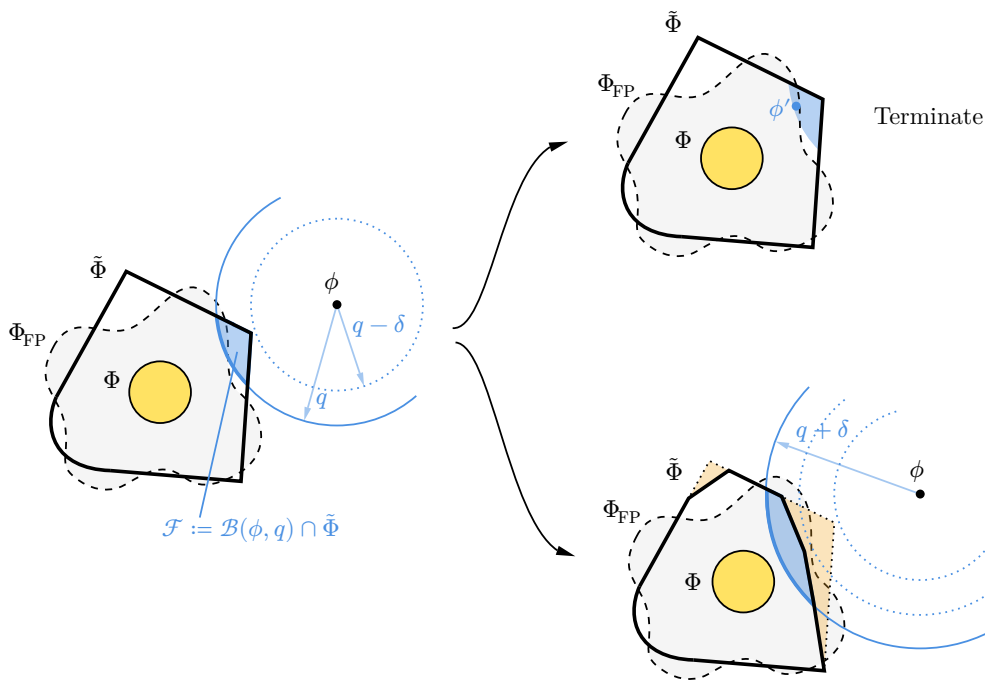


Figure 1: Taken from Daskalakis, Farina, Fishelson, Pipis and Schneider [DFF+25]. The set of linear endomorphisms on \mathcal{X} is denoted by Φ . Φ_{FP} is the set of transformations containing a fixed point in \mathcal{X} . Clearly $\Phi \subseteq \Phi_{\text{FP}}$. The algorithm operates over a changing “shell set” $\tilde{\Phi}$ which, unlike Φ , admits an efficient description. Given a function φ , the key idea is that if φ does not admit a fixed point in \mathcal{X} , one can shrink $\tilde{\Phi}$ by a significant margin. Otherwise, the algorithm can safely output a fixed point of φ .

Bibliography for this chapter

- [ZAT+25a] B. H. Zhang, I. Anagnostides, E. Tewolde, R. E. Berker, G. Farina, V. Conitzer and T. Sandholm. (2025). Learning and Computation of Φ -Equilibria at the Frontier of Tractability. *Economics and Computation*.

- [ZAFS24] B. H. Zhang, I. Anagnostides, G. Farina and T. Sandholm. (2024). Efficient Φ -Regret Minimization with Low-Degree Swap Deviations in Extensive-Form Games. *Advances in Neural Information Processing Systems*.
- [DFF+25] C. Daskalakis, G. Farina, M. Fishelson, C. Pipis and J. Schneider. (2025). Efficient Learning and Computation of Linear Correlated Equilibrium in General Convex Games. *Symposium on Theory of Computing (STOC)*.
- [ZAT+25b] B. H. Zhang, I. Anagnostides, E. Tewolde, R. E. Berker, G. Farina, V. Conitzer and T. Sandholm. (2025). Expected Variational Inequalities. *International Conference on Machine Learning*, 74422–74446.
- [ZFS24] B. H. Zhang, G. Farina and T. Sandholm. (2024). Mediator Interpretation and Faster Learning Algorithms for Linear Correlated Equilibria in General Sequential Games. *International Conference on Learning Representations*.