

Chapter 6

Profile Swap Regret, Manipulability, and Response-Based Approachability

Ioannis Anagnostides, Gabriele Farina, and Brian Hu Zhang*

In this chapter, we introduce the notion of *profile swap regret*, which lies in between linear swap regret and full swap regret. A key property of profile swap regret is that it guarantees *non-manipulability*—in a sense that will be formalized soon—against a dynamic optimizer. What’s more, there is an efficient algorithm for minimizing profile swap regret. This notion was recently introduced by [Arunachaleswaran, Collina, Mansour, Mohri, Schneider and Sivan \[ACM+25\]](#), who also provided the first efficient algorithm. In what follows, we present the approach of [Anagnostides, Farina, Fishelson, Luo and Schneider \[AFF+26\]](#), which in turn relies on the response-based approachability algorithm of [Bernstein and Shimkin \[BS15\]](#).¹

6.1 Setup

We operate in the usual online linear optimization setting. The learner picks strategies from a convex and compact strategy set $\mathcal{X} \subset \mathbb{R}^d$. At every round $t \in [T]$, the learner first selects a strategy $\mathbf{x}^{(t)} \in \mathcal{X}$; the adversary selects a utility vector $\mathbf{u}^{(t)} \in \mathcal{U} \subseteq \mathbb{R}^d$; then the learner receives utility $\langle \mathbf{x}^{(t)}, \mathbf{u}^{(t)} \rangle$ and observes $\mathbf{u}^{(t)}$ as feedback. We make the standard normalization assumption $|\langle \mathbf{x}, \mathbf{u} \rangle| \leq 1$ for all $\mathbf{x} \in \mathcal{X}, \mathbf{u} \in \mathcal{U}$. In what follows, for a utility vector $\mathbf{u} \in \mathcal{U}$, we define the best-response map $b(\mathbf{u}) \in \arg \max_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{x}, \mathbf{u} \rangle$, with ties broken arbitrarily.

Linear swap regret. We first recall the benchmark of linear swap regret. An affine map $\varphi : \mathcal{X} \rightarrow \mathbb{R}^d$ is an *endomorphism* of \mathcal{X} if $\varphi(\mathcal{X}) \subseteq \mathcal{X}$. We denote by $\text{End}(\mathcal{X})$ the set of affine endomorphisms. We write each $\varphi \in \text{End}(\mathcal{X})$ as $\varphi(\mathbf{x}) = \mathbf{M}\mathbf{x} + \mathbf{a}$. The *linear swap regret* of the learner after T rounds is

$$\text{LinearSwapReg}^{(T)} := \max_{\varphi \in \text{End}(\mathcal{X})} \sum_{t=1}^T \langle \varphi(\mathbf{x}^{(t)}), \mathbf{u}^{(t)} \rangle - \sum_{t=1}^T \langle \mathbf{x}^{(t)}, \mathbf{u}^{(t)} \rangle.$$

This is a natural relaxation of (full) swap regret: instead of allowing arbitrary maps, the performance of the learner is compared against affine endomorphisms. In this case, the algorithm

*✉ 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.

¹We point the interested reader to <https://sites.google.com/view/strategic-learning-ec25> for an adjacent workshop at EC '25.

of Gordon, Greenwald and Marks [GGM08] reduces linear swap regret to external regret over the $d(d+1)$ -dimensional set of deviations. This implies an online algorithm with linear swap regret growing as $O(d\sqrt{T})$, which is information-theoretically optimal [AFF+26]. On the other hand, the algorithm of Gordon, Greenwald and Marks [GGM08] is computationally inefficient [DFF+25]. This lecture will cover an approachability framework that circumvents this computational barrier.

Correlated strategy profiles. A basic object in the approachability formulation is the *correlated strategy profile (CSP)*. Specifically, for a realized pair $(\mathbf{x}^{(t)}, \mathbf{u}^{(t)})$, we define $\kappa^{(t)} := (\mathbf{u}^{(t)} \otimes \mathbf{x}^{(t)}, \mathbf{u}^{(t)}) \in \mathbb{R}^{d \times (d+1)}$. The average CSP is

$$\bar{\kappa}^{(T)} := \frac{1}{T} \sum_{t=1}^T \kappa^{(t)}. \quad (1)$$

For a CSP $\kappa = (\mathbf{K}, \mathbf{q})$ and an affine map $\varphi(\mathbf{x}) = \mathbf{M}\mathbf{x} + \mathbf{a}$, we use the shorthand notation $\langle (\mathbf{M}, \mathbf{a}), \kappa \rangle := \langle \mathbf{M}, \mathbf{K} \rangle + \langle \mathbf{a}, \mathbf{q} \rangle$. With this notation at hand, linear swap regret can be rewritten as

$$\text{LinearSwapReg}^{(T)} = T \max_{\varphi=(\mathbf{M}, \mathbf{a}) \in \text{End}(\mathcal{X})} \langle (\mathbf{M}, \mathbf{a}) - (\mathbf{I}, 0), \bar{\kappa}^{(T)} \rangle. \quad (2)$$

Using this reformulation, we will shortly see how to reduce minimizing linear swap regret to a suitable approachability problem.

6.2 Profile swap regret and non-manipulability

We now define the notion of *profile swap regret* and discuss its non-manipulability properties. To do so, we define the instantaneous regret as $R(\mathbf{x}, \mathbf{u}) := \max_{\mathbf{x}^* \in \mathcal{X}} \langle \mathbf{x}^* - \mathbf{x}, \mathbf{u} \rangle$. Equivalently, we can write $R(\mathbf{x}, \mathbf{u}) = \langle b(\mathbf{u}) - \mathbf{x}, \mathbf{u} \rangle$. A valid decomposition of $\bar{\kappa}^{(T)}$ can be written as

$$\bar{\kappa}^{(T)} = \sum_{j=1}^m \lambda_j (\mathbf{u}_j \otimes \mathbf{x}_j),$$

where $\lambda_j \geq 0$, $\sum_{j=1}^m \lambda_j = 1$, $\mathbf{x}_j \in \mathcal{X}$, and $\mathbf{u}_j \in \mathcal{U}$.

Definition 6.1 (Profile swap regret). The *profile swap regret* of a CSP $\bar{\kappa}^{(T)}$ is

$$\text{ProfileSwapReg}^{(T)}(\bar{\kappa}^{(T)}) := T \min_{(\lambda_j, \mathbf{x}_j, \mathbf{u}_j)_j} \sum_j \lambda_j R(\mathbf{x}_j, \mathbf{u}_j), \quad (3)$$

where the minimum is over all valid convex decompositions of $\bar{\kappa}^{(T)}$.

The definition is admittedly somewhat cumbersome, but as we shall see it unlocks certain powerful properties. This is a good time to discuss the notion of non-manipulability due to Deng, Schneider and Sivan [DSS19].

6.2.1 Manipulability against a dynamic optimizer

We consider a two-player game between a *learner* and an *optimizer*. The learner follows some learning algorithm, whereas the optimizer may choose its actions strategically over time after observing how the learner updates. We let u_ℓ be the learner’s utility function and \mathcal{X} its strategy set. Similarly, we let u_o be the optimizer’s utility function and \mathcal{Y} its strategy set.

The key relevant benchmark in this setting is the optimizer’s *Stackelberg value*, which is the best utility the optimizer can guarantee by committing to a fixed strategy $\mathbf{y} \in \mathcal{Y}$, while the learner best-responds to that strategy (with ties broken in favor of the optimizer):

$$\text{Stack}(u_o) := \max_{\mathbf{y} \in \mathcal{Y}} \max_{\mathbf{x} \in \text{BR}(\mathbf{y})} u_o(\mathbf{x}, \mathbf{y}),$$

where $\text{BR}(\mathbf{y}) := \arg \max_{\mathbf{x} \in \mathcal{X}} u_\ell(\mathbf{x}, \mathbf{y})$. The Stackelberg value is what the optimizer can guarantee by means of a *static* commitment. Informally speaking, a learning algorithm is said to be manipulable if the optimizer can do strictly better than its Stackelberg value by using a *dynamic* strategy: instead of playing a fixed strategy \mathbf{y} , the optimizer chooses a sequence $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots$ so as to steer the learner’s future play toward more favorable outcomes. (It’s interesting to note that this steering problem can be computationally intractable [ADD24].) Equivalently, even if the optimizer fully understands the learner’s algorithm and chooses its strategies dynamically to influence future play, it cannot asymptotically obtain more than what it could already obtain by committing to a fixed strategy.

A beautiful connection crystallized by Arunachaleswaran, Collina, Mansour, Mohri, Schneider and Sivan [ACM+25] reassures us that a learning algorithm that has vanishing profile swap regret cannot be manipulated by a dynamic optimizer.

Proposition 6.2. If a learning algorithm guarantees $\text{ProfileSwapReg}^{(T)} = o(T)$, it is non-manipulable.

6.3 Reducing to an approachability problem

Having introduced and motivated the notion of profile swap regret, we now reduce it to an approachability instance. We define $\mathcal{K} := \text{conv}\{(\mathbf{u} \otimes \mathbf{x}, \mathbf{u}) : \mathbf{u} \in \mathcal{U}, \mathbf{x} \in \mathcal{X}\}$ and $\mathcal{S} := \text{conv}\{(\mathbf{u} \otimes b(\mathbf{u}), \mathbf{u}) : \mathbf{u} \in \mathcal{U}\}$. We measure approachability loss by

$$\text{AppLoss}^{(T)} := \min_{\mathbf{s} \in \mathcal{S}} \|\bar{\kappa}^{(T)} - \mathbf{s}\|_F. \quad (4)$$

We first point out that minimizing linear swap regret reduces to this approachability instance.

Lemma 6.3 ([AFF+26]). For any time $T \in \mathbb{N}$,

$$\text{LinearSwapReg}^{(T)} \leq 2T \text{AppLoss}^{(T)} \cdot \max_{\varphi \in \text{End}(\mathcal{X})} \|\varphi\|_F. \quad (5)$$

Proof. Let $\mathbf{s} \in \mathcal{S}$ be closest to $\bar{\kappa}^{(T)}$. Since \mathbf{s} is, by definition, a convex combination of points of the form $(\mathbf{u} \otimes b(\mathbf{u}), \mathbf{u})$ and $b(\mathbf{u})$ maximizes $\langle \cdot, \mathbf{u} \rangle$ over \mathcal{X} , every endomorphism $\varphi \in \text{End}(\mathcal{X})$ satisfies

$$\langle \varphi - (\mathbf{I}, \mathbf{0}), \mathbf{s} \rangle \leq 0.$$

Therefore, using (2),

$$\text{LinearSwapReg}^{(T)} \leq T \max_{\varphi \in \text{End}(\mathcal{X})} \langle \varphi - (\mathbf{I}, \mathbf{0}), \bar{\kappa}^{(T)} - \mathbf{s} \rangle.$$

The claim now follows from Cauchy-Schwarz. \square

What's more, the same target set also captures profile swap regret. The following lemma follows essentially by the definition of profile swap regret.

Lemma 6.4 ([AFF+26]). The profile swap distance of $\bar{\kappa}^{(T)}$ is equal to $\text{AppLoss}^{(T)}$.

The profile swap distance of a CSP $\bar{\kappa}^{(T)}$ is the infimum Euclidean distance of $\bar{\kappa}^{(T)}$ from a CSP with zero profile swap regret. As a result, we conclude that minimizing the approachability loss simultaneously circumscribes both linear swap regret and profile swap regret.

6.4 Response-based approachability

The key observation now is that the induced approachability problem can be solved efficiently through the response-based approachability framework of Bernstein and Shimkin [BS15]. This was recently leveraged by Anagnostides, Farina, Fishelson, Luo and Schneider [AFF+26].

Algorithm [BS15]: Response-based approachability

Input: Horizon T , sets \mathcal{X}, \mathcal{U} , best-response map $b : \mathcal{U} \rightarrow \mathcal{X}$

Initialize $\mathbf{U}^{(0)} := \mathbf{0} \in \mathbb{R}^{d \times (d+1)}$

for each round $t = 1, \dots, T$ **do**

Compute maximin strategies $(\mathbf{x}^{(t)}, \mathbf{u}_*^{(t)})$ for

$$\max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{u} \in \mathcal{U}} \langle \mathbf{U}^{(t-1)}, (\mathbf{u} \otimes \mathbf{x}, \mathbf{u}) \rangle$$

Set $\mathbf{s}^{(t)} := (\mathbf{u}_*^{(t)} \otimes b(\mathbf{u}_*^{(t)}), \mathbf{u}_*^{(t)})$

Play $\mathbf{x}^{(t)}$ and observe $\mathbf{u}^{(t)}$

Set $\kappa^{(t)} := (\mathbf{u}^{(t)} \otimes \mathbf{x}^{(t)}, \mathbf{u}^{(t)})$

Update $\mathbf{U}^{(t)} := \mathbf{U}^{(t-1)} + \kappa^{(t)} - \mathbf{s}^{(t)}$

The algorithm proceeds by maintaining the accumulated displacement

$$\mathbf{U}^{(t)} := \sum_{\tau=1}^t (\kappa^{(\tau)} - \mathbf{s}^{(\tau)}) \in \mathbb{R}^{d \times (d+1)}.$$

At every round t , it solves the bilinear zero-sum game

$$\max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{u} \in \mathcal{U}} \langle \mathbf{U}^{(t-1)}, (\mathbf{u} \otimes \mathbf{x}, \mathbf{u}) \rangle. \quad (6)$$

Let $(\mathbf{x}^{(t)}, \mathbf{u}_*^{(t)})$ be a pair of minimax strategies for this game. The algorithm then sets

$$\mathbf{s}^{(t)} := (\mathbf{u}_*^{(t)} \otimes b(\mathbf{u}_*^{(t)}), \mathbf{u}_*^{(t)}) \in \mathcal{S}.$$

Finally, it plays the strategy $\mathbf{x}^{(t)}$, whereupon it observes the utility $\mathbf{u}^{(t)}$, forms the induced $\kappa^{(t)}$, and updates the cumulative displacement $\mathbf{U}^{(t)}$. All steps can be efficiently implemented through oracle access to \mathcal{X} .

The proof of correctness crucially relies on the minimax theorem, as we formalize below.

Lemma 6.5 ([BS15]). For every utility $\mathbf{u}^{(t)} \in \mathcal{U}$, we have

$$\langle \mathbf{U}^{(t-1)}, \kappa^{(t)} - \mathbf{s}^{(t)} \rangle \geq 0. \quad (7)$$

Proof. Let

$$g_t(\mathbf{x}, \mathbf{u}) := \langle \mathbf{U}^{(t-1)}, (\mathbf{u} \otimes \mathbf{x}, \mathbf{u}) \rangle.$$

By the minimax theorem,

$$\max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{u} \in \mathcal{U}} g_t(\mathbf{x}, \mathbf{u}) = \min_{\mathbf{u} \in \mathcal{U}} \max_{\mathbf{x} \in \mathcal{X}} g_t(\mathbf{x}, \mathbf{u}).$$

Thus,

$$g_t(\mathbf{x}^{(t)}, \mathbf{u}^{(t)}) \geq \min_{\mathbf{u} \in \mathcal{U}} \max_{\mathbf{x} \in \mathcal{X}} g_t(\mathbf{x}, \mathbf{u}).$$

Since $\mathbf{u}_*^{(t)}$ is minimax optimal and $b(\mathbf{u}_*^{(t)})$ is a feasible strategy in \mathcal{X} ,

$$\min_{\mathbf{u} \in \mathcal{U}} \max_{\mathbf{x} \in \mathcal{X}} g_t(\mathbf{x}, \mathbf{u}) \geq g_t(b(\mathbf{u}_*^{(t)}), \mathbf{u}_*^{(t)}).$$

Combining the two inequalities gives

$$\langle \mathbf{U}^{(t-1)}, \kappa^{(t)} \rangle \geq \langle \mathbf{U}^{(t-1)}, \mathbf{s}^{(t)} \rangle,$$

which is exactly (7). □

The invariant implies approachability by the usual Pythagorean argument.

Lemma 6.6 (Pythagorean lemma). Let $\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(T)}$ be vectors with $\|\mathbf{z}^{(t)}\|_2 \leq B$ for all t . If

$$\left\langle \sum_{\tau=1}^{t-1} \mathbf{z}^{(\tau)}, \mathbf{z}^{(t)} \right\rangle \leq 0$$

for every t , then

$$\left\| \sum_{t=1}^T \mathbf{z}^{(t)} \right\|_2 \leq B\sqrt{T}.$$

Applying Lemma 6.6 with $\mathbf{z}^{(t)} = \mathbf{s}^{(t)} - \boldsymbol{\kappa}^{(t)}$ gives

$$\text{AppLoss}^{(T)} \leq \left\| \frac{1}{T} \sum_{t=1}^T (\boldsymbol{\kappa}^{(t)} - \mathbf{s}^{(t)}) \right\|_F \leq \frac{2B}{\sqrt{T}}, \quad (8)$$

where $B := \max_{\mathbf{x} \in \mathcal{X}, \mathbf{u} \in \mathcal{U}} \|(\mathbf{u} \otimes \mathbf{x}, \mathbf{u})\|_F$. We thus arrive at the following theorem.

Theorem 6.7 ([AFF+26]). For any sequence of utilities $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(T)} \in \mathcal{U}$, the response-based approachability algorithm guarantees

$$\text{LinearSwapReg}^{(T)} \leq 4\sqrt{T} \left(\max_{\mathbf{x} \in \mathcal{X}, \mathbf{u} \in \mathcal{U}} \|\mathbf{u}\|_2 \sqrt{\|\mathbf{x}\|_2^2 + 1} \right) \left(\max_{\varphi \in \text{End}(\mathcal{X})} \|\varphi\|_F \right), \quad (9)$$

and

$$\text{ProfileSwapDist}^{(T)} \leq \frac{2}{\sqrt{T}} \left(\max_{\mathbf{x} \in \mathcal{X}, \mathbf{u} \in \mathcal{U}} \|\mathbf{u}\|_2 \sqrt{\|\mathbf{x}\|_2^2 + 1} \right). \quad (10)$$

Through suitable preconditioning, it can be shown that this algorithm guarantees optimal linear swap regret [AFF+26]. We further remark that this response-based framework can be extended to nonlinear deviations; we refer the interested reader to [Anagnostides, Farina, Fishelson, Luo and Schneider \[AFF+26\]](#).

Bibliography for this chapter

- [ACM+25] E. R. Arunachaleswaran, N. Collina, Y. Mansour, M. Mohri, J. Schneider and B. Sivan. (2025). Swap Regret and Correlated Equilibria Beyond Normal-Form Games. *Conference on Economics and Computation (EC)*.
- [AFF+26] I. Anagnostides, G. Farina, M. Fishelson, H. Luo and J. Schneider. (2026). Swap Regret Minimization Through Response-Based Approachability. *Conference on Learning Theory (COLT)*.
- [BS15] A. Bernstein and N. Shimkin. (2015). Response-Based Approachability with Applications to Generalized No-Regret Problems. *Journal of Machine Learning Research*, 16, 747–773.
- [GGM08] G. J. Gordon, A. Greenwald and C. Marks. (2008). No-regret learning in convex games. *International Conference on Machine Learning*.
- [DFE+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)*.
- [DSS19] Y. Deng, J. Schneider and B. Sivan. (2019). Strategizing against no-regret learners. *Neural information processing systems (NeurIPS)*.
- [ADD24] A. Assos, Y. Dagan and C. Daskalakis. (2024). Maximizing utility in multi-agent environments by anticipating the behavior of other learners. *Neural Information Processing Systems (NeurIPS)*.