Pareto-Optimal Algorithms for Learning in Repeated Games

Eshwar Ram Arunachaleswaran
University of Pennsylvania
Visiting the Simons Institute

Natalie Collina
University of Pennsylvania

Jon Schneider
Google Research

Talk at Georgia Tech

Based on work that appeared at EC 2024

Motivation

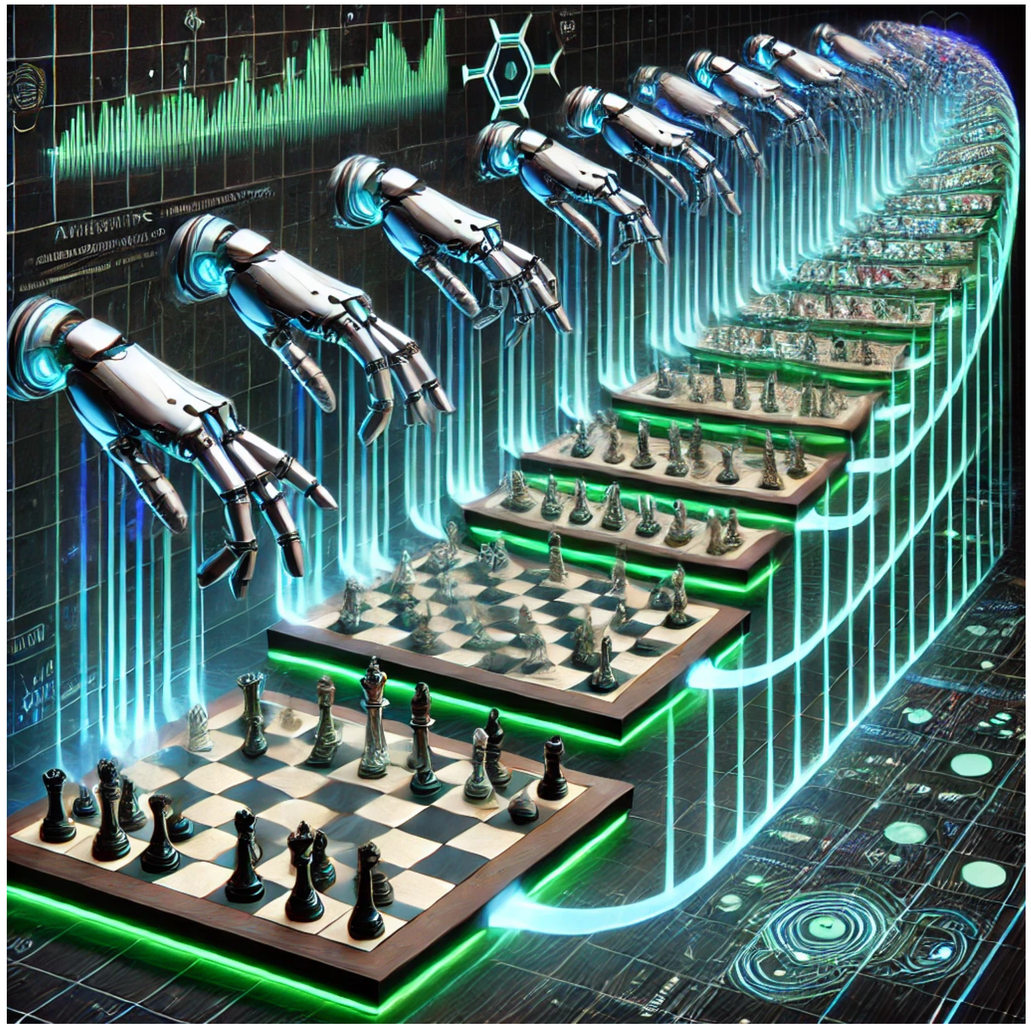
Algorithms are used to make Decisions in Strategic Environments

Pricing Algorithms

Automated Bidding

Motivation

Algorithms are Strategies in Repeated Games



What makes learning to play games difficult (and interesting)?

- Adaptive and temporal strategies of the other player(s) makes the environment non-stationary.
- Without knowing the nature of the other players, impossible to find the optimal algorithm/ policy.

What makes learning to play games more tractable than arbitrary adversarial environments?

- Opponents behavior might reveal their nature.
- With rational opponents, the environment is not arbitrary since they are consistently optimizing for some (potentially unknown) objective.

Algorithms as Strategies

For Repeated Games

Scenario : Algorithms for Repeated Games

Theory CS Answer : No-Regret Algorithms!

Fact : All players playing no-(swap)-regret leads to convergence on average to (Coarse) Correlated Equilibria

Some Issues and Questions

- Fact: No-Regret is not always a best response to No-Regret.
- What does an (algorithmic) best-response to no-regret look like?
- What behavior can interaction between algorithms induce?

Central Question : No-regret algorithms (and variants) represent the best we can do for arbitrary online environments. Can we ask for stronger properties in repeated games?

Sneak Peek

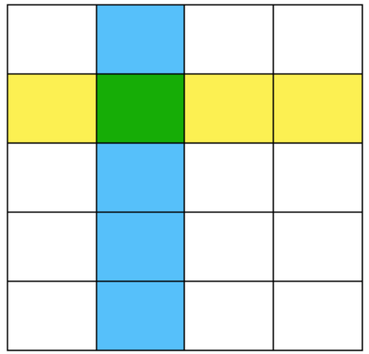
- Pareto-Optimality
- Non-Manipulability

The Model

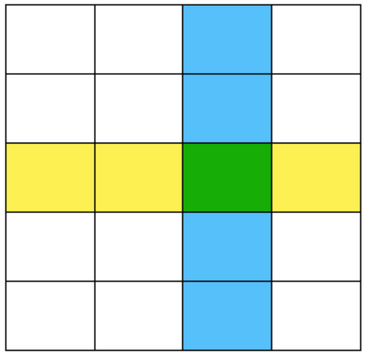
Learners and Optimizers

Repeated Games

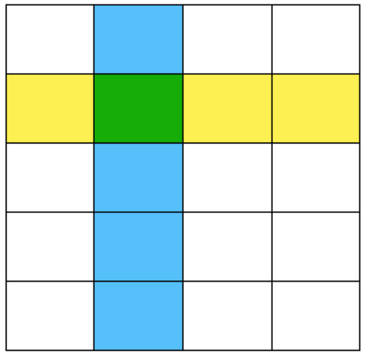
The action space is algorithms



Round 1

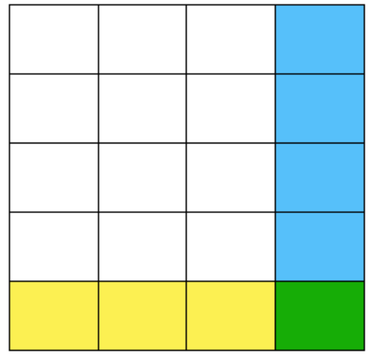


Round 2



Round 3

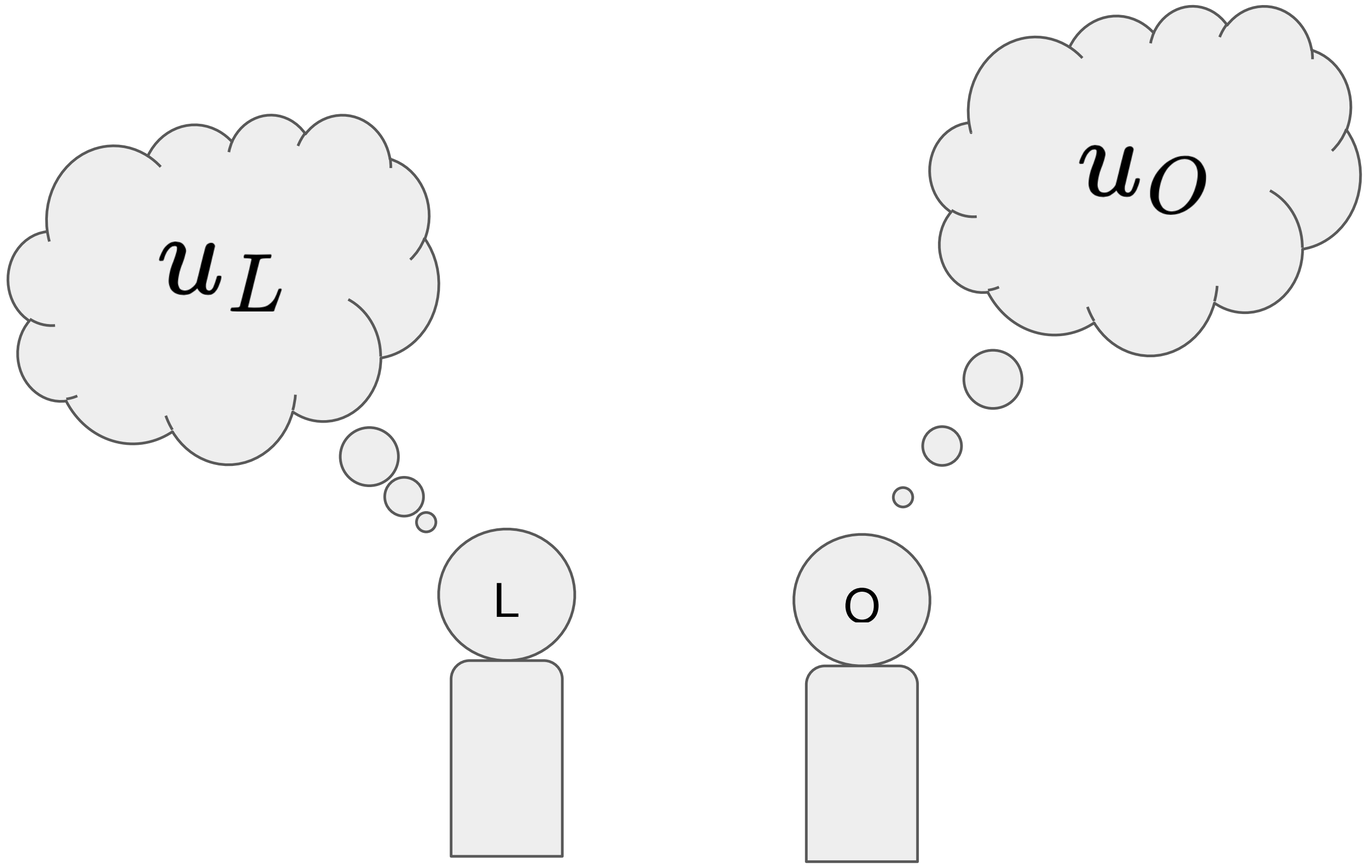
...



Round T

Can be adaptive and time-dynamic

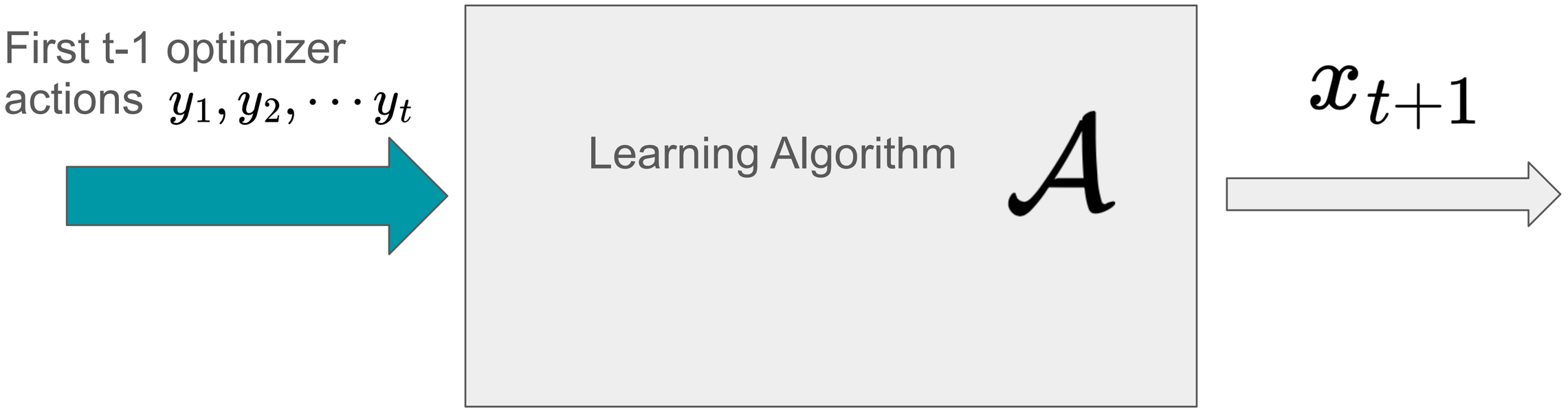
Two Player Repeated Games



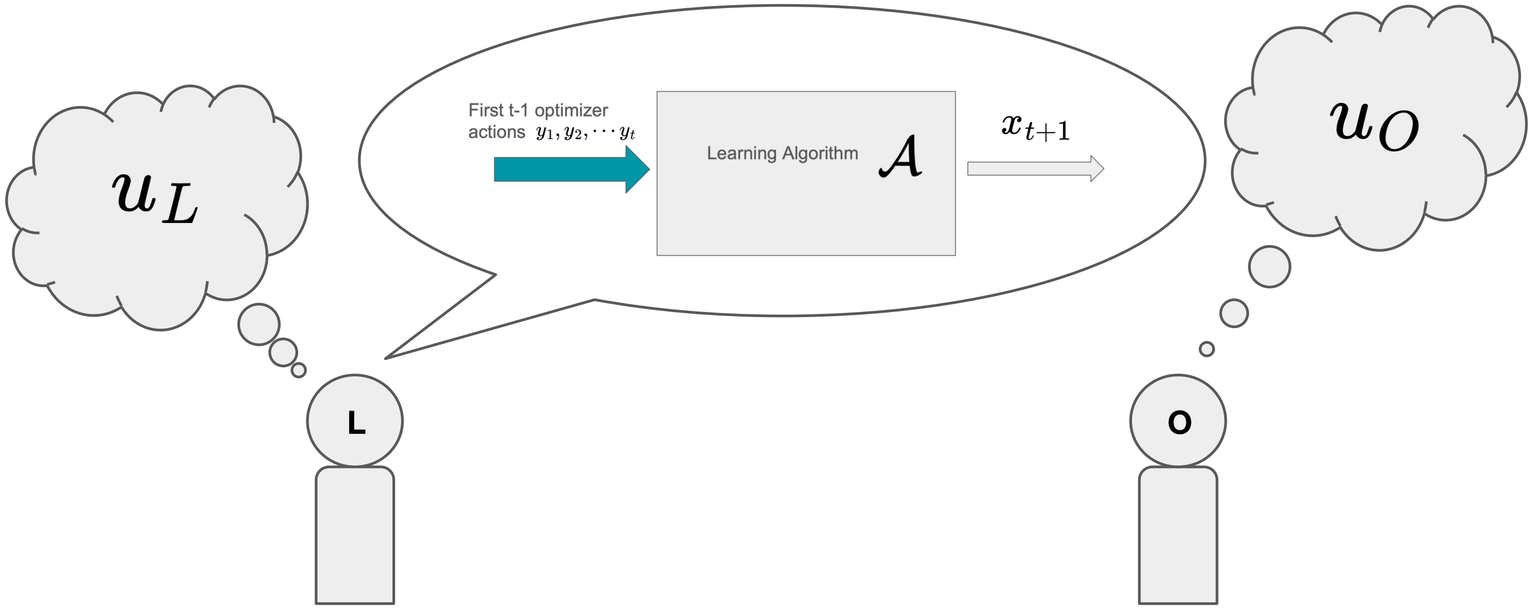
Two player asymmetric setup with one player called the “learner” and the other called the “optimizer”

- The Learner has an action set Δ_m
- The Optimizer has an action set Δ_n
- They simultaneously play actions x_t, y_t in the t-th round and then observe the other player’s action.
- Bilinear utility functions u_L, u_O

The Learner Algorithm

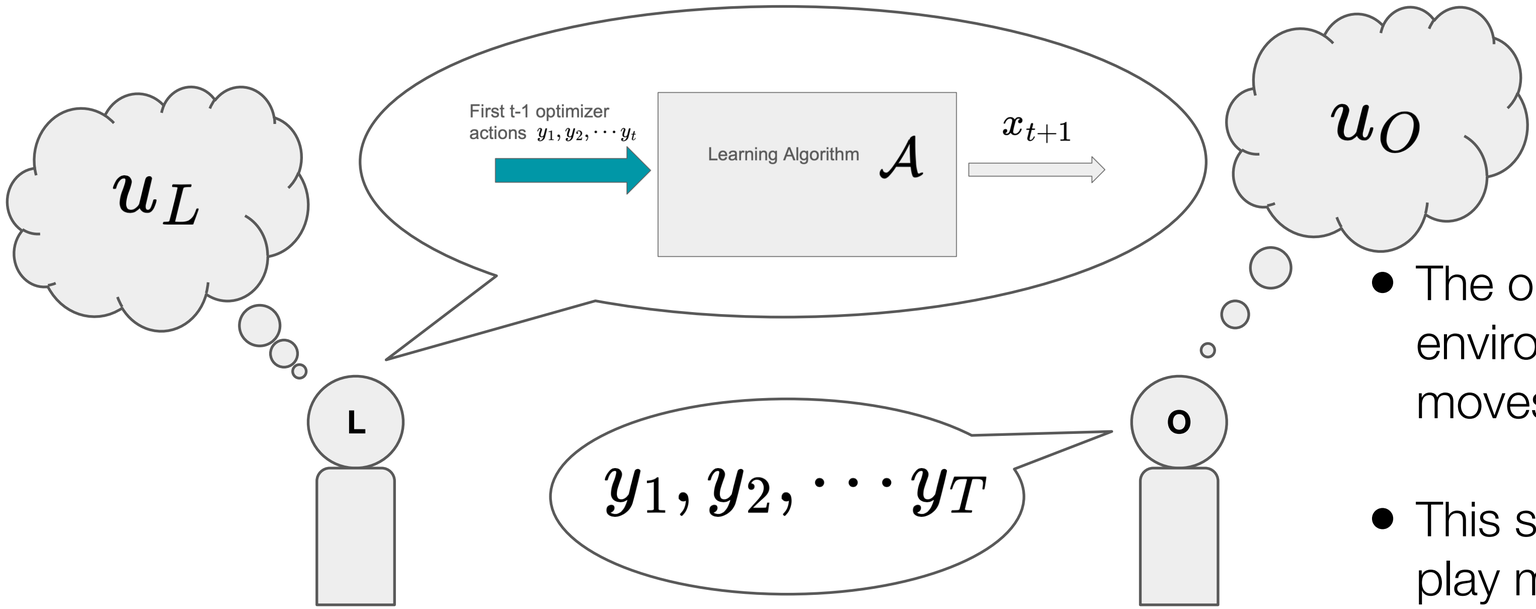


The Learner Algorithm



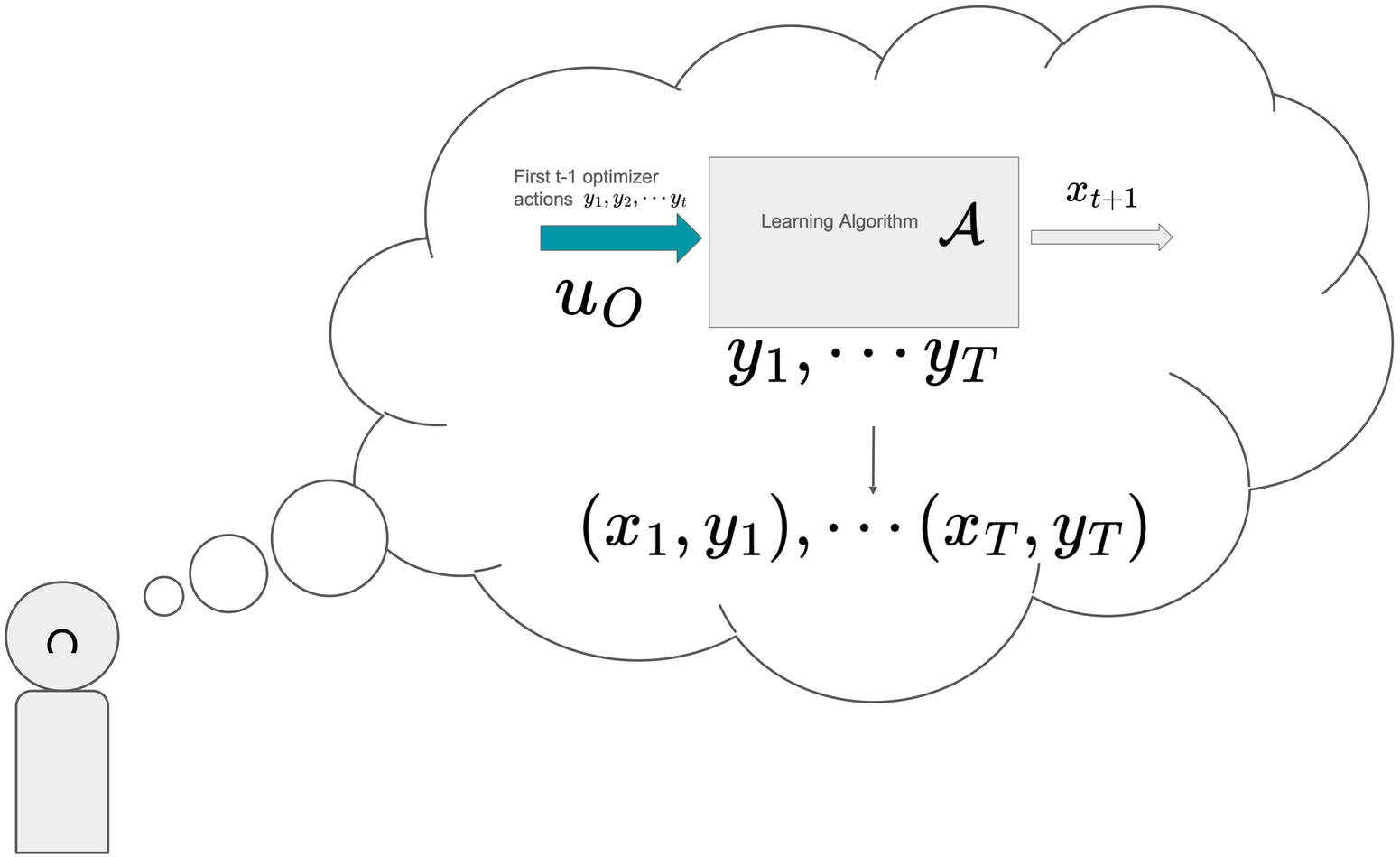
The learner publicly commits to a learning algorithm

The Optimizer's Role



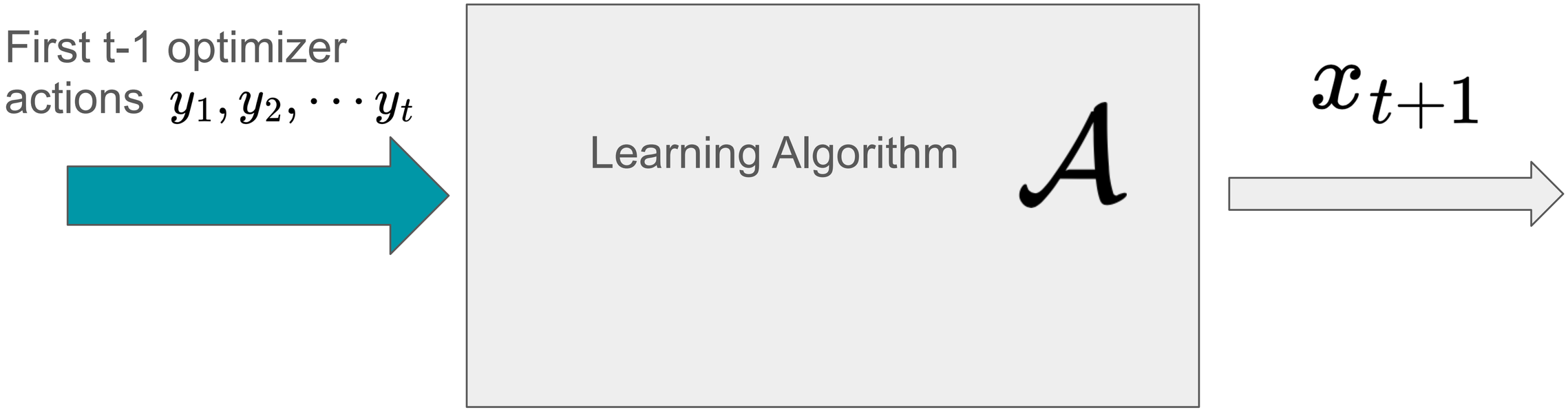
- The optimizer has full information about the environment and plays the optimal sequence of moves to maximize their private payoff
- This sequence induces a particular transcript of play maximizing the optimizer payoff and thus also fixes the learner's payoff

Best-responses to Algorithms

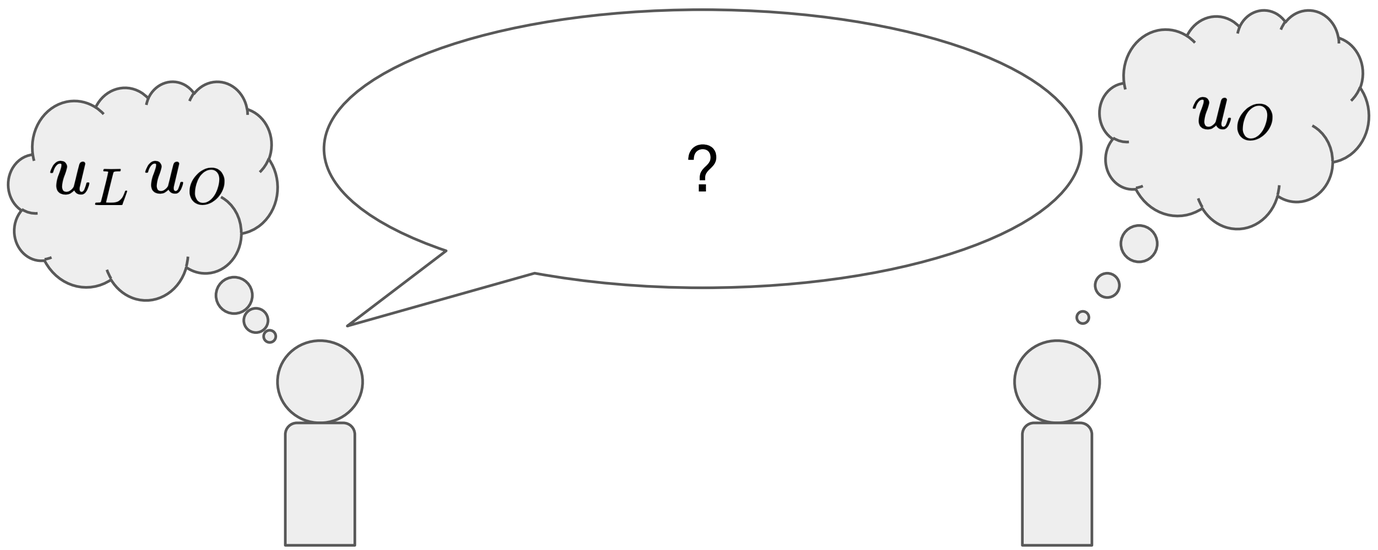


Our Question

Given this setup, what learning algorithm should the learner commit to to maximize *their own* (limit average) payoff ?



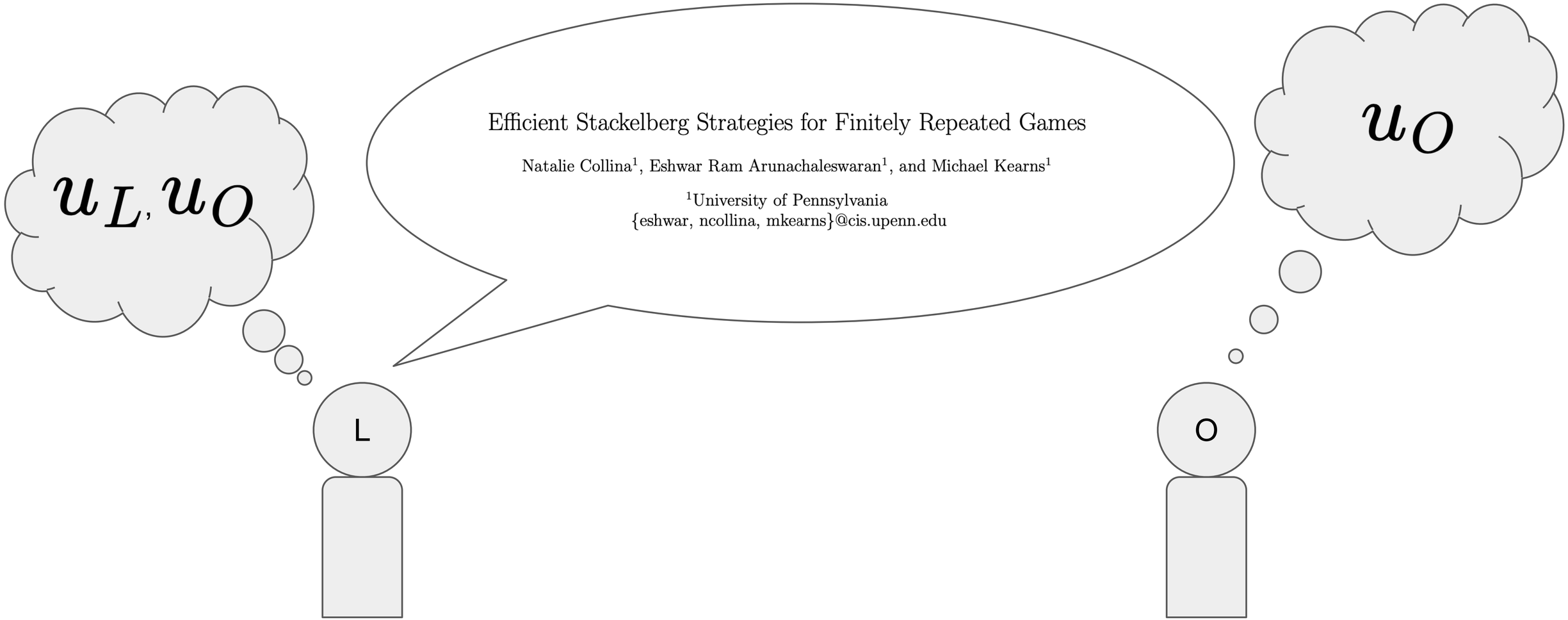
Full-information Setting



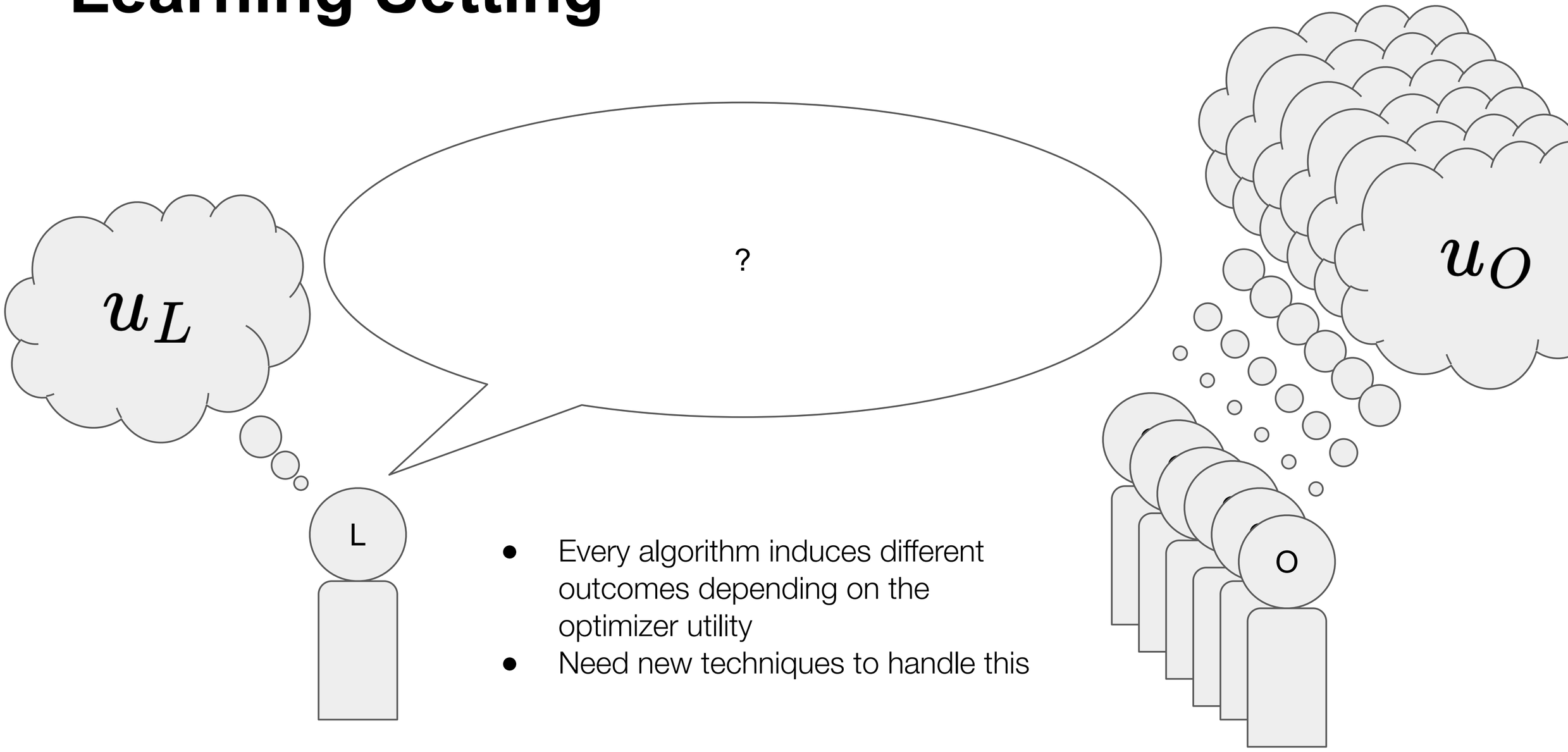
- Knowing the optimizer's payoff upfront implies the existence of a well defined bilevel optimization problem for the learner.
- In particular, the learner can assess the payoff associated with any commitment by simulating a rational optimizer response.

The Stackelberg Equilibrium Problem

With algorithms as strategies



Learning Setting



- Every algorithm induces different outcomes depending on the optimizer utility
- Need new techniques to handle this

What are good learning algorithms to use?

Optimistic: Pointwise optimality against all optimizers 

Our answer: No-Regret + **Pareto-Optimality**

Standard benchmark: No-Regret on every transcript

Multiplicative weights, etc.

No-Regret Algorithms

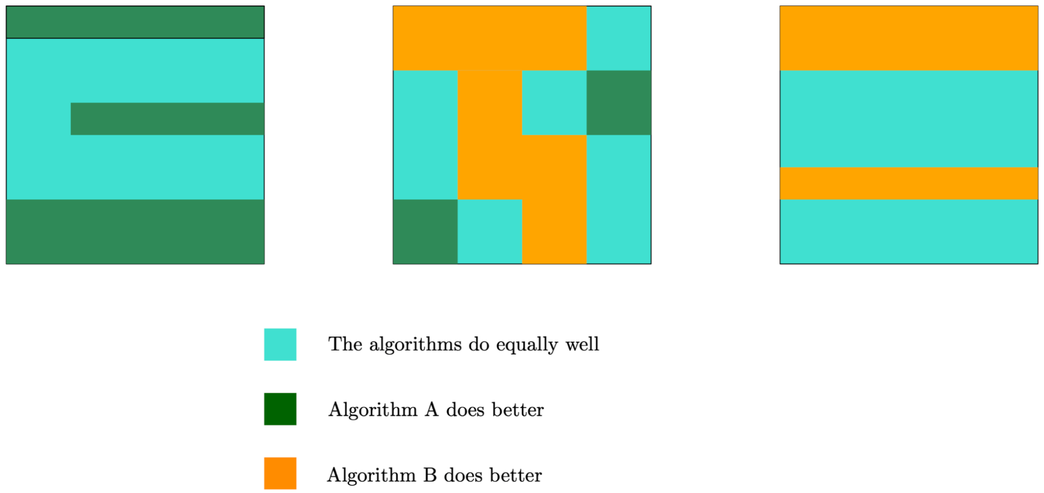
$$\sum_{t=1}^{T} u_L(x_t, y_t) \geq \left(\max_{y^* \in [n]} \sum_{t=1}^{T} u_L(x_t, y^*) \right) - o(T).$$

Competitive with the best fixed action in hindsight on every transcript

- Existence implied by the minimax theorem.
- Multiple Algorithms exist — including multiplicative weights, Follow-the-Perturbed-Leader, Gradient Descent, etc.
- Efficient algorithms exist even for some problems with large action spaces — such as online shortest path

Pareto-Optimality

Comparing Algorithms : Three Possible Outcomes

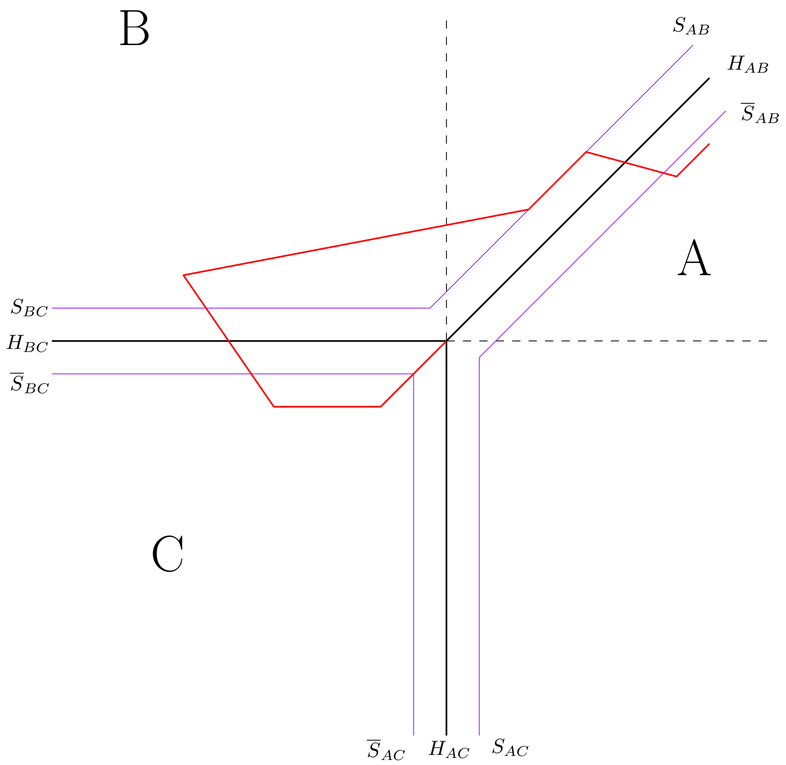


- Each learning algorithm has an associated infinite profile of payoffs measured against all possible optimizers.
- A Pareto-Optimal learning algorithm has a payoff profile on the Pareto-frontier.
- All other algorithms are Pareto-dominated, and thus there is a natural reason to never use such algorithms.

Question: Does a non-trivial Pareto-dominated algorithm exist?

Result 1 : Popular No-Regret Algorithms are Pareto-Dominated

Theorem 1 : All Follow-the-Regularized-Leader algorithms, which include Multiplicative Weights, FTPL, Lazy Gradient Descent, are Pareto-dominated.



- Results based on a (partial) characterization of best-response sequences against FTRL algorithms.
- These algorithms have lacunae (that can be eliminated) due to a property they share, called the “mean-based” property.
- The dominating algorithms do not necessarily have a succinct representation.

Result 2 : No-Swap-Regret Algorithms are Pareto-Optimal

Theorem 2 : All No-Swap-Regret algorithms are Pareto-Optimal. Additionally, all no-swap-regret algorithms are strategically equivalent in terms of the limit average payoff.

$$\sum_{t=1}^{T} u_L(x_t, y_t) \geq \max_{\pi:[n]\to[n]} \sum_{t=1}^{T} u_L(x_t, \pi(y_t)) - o(T).$$

Competitive with the best fixed action on the **subsequence** that action i is played, for all i in [n].

- NSR is a strengthening of the no-regret property.
- First explicit construction by Blum and Mansour (06).
- Recent results (DDFG 24 and PR 24) show fundamentally different constructions that reduce no-swap-regret to no-regret.

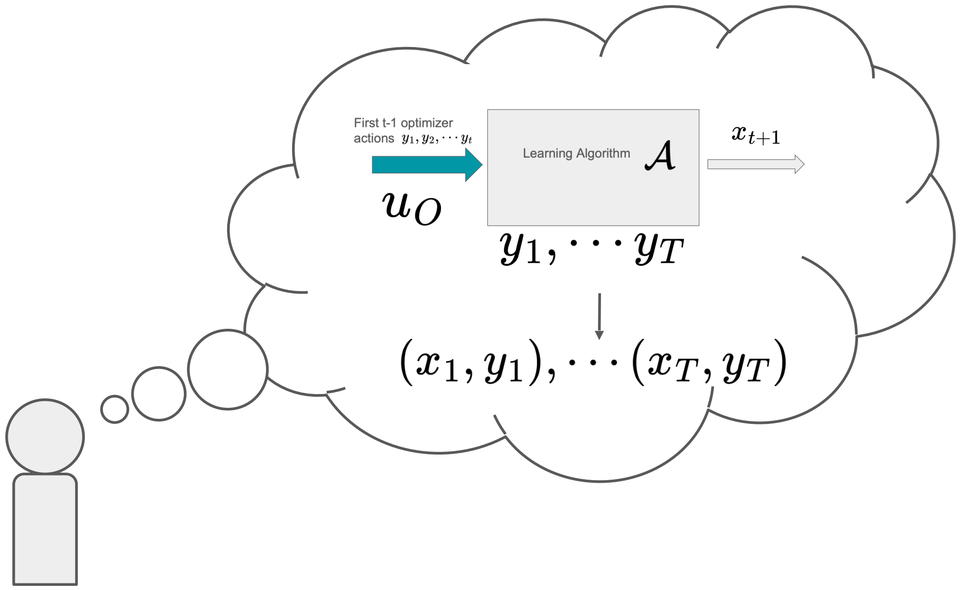
Related Work

Non-Manipulability

Learning Algorithms for Repeated Games

1. Brown, W., Schneider, J., & Vodrahalli, K. (2024). Is learning in games good for the learners? Advances in Neural Information Processing Systems, 36.
2. Deng, Y., Schneider, J., & Sivan, B. (2019). Strategizing against no-regret learners. Advances in Neural Information Processing Systems, 32.
3. Collina, N., Arunachaleswaran, E. R., & Kearns, M. (2023). Efficient Stackelberg strategies for finitely repeated games. Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, 643–651.
4. Haghtalab, N., Lykouris, T., Nietert, S., & Wei, A. (2022). Learning in Stackelberg games with non-myopic agents. Proceedings of the 23rd ACM Conference on Economics and Computation, 917–918.
5. Mansour, Y., Mohri, M., Schneider, J., & Sivan, B. (2022). Strategizing against learners in Bayesian games. Conference on Learning Theory, 5221–5252.
6. Kumar, R., Schneider, J., & Sivan, B. (2024). Strategically-robust learning algorithms for bidding in first-price auctions. ACM Conference on Economics and Computation
7. Rubinstein, A., & Zhao, J. (2024). Strategizing against no-regret learners in first-price auctions. ACM Conference on Economics and Computation
8. Brânzei, S., Hajiaghayi, M. T., Phillips, R., Shin, S., & Wang, K. (2024). Dueling over dessert, mastering the art of repeated cake cutting, Advances in Neural Information Processing Systems, 36.

Non-Manipulability



A learning algorithm is non-manipulable if the optimizer has a best-response that is static over time.

Known Results — DSS 19 show that

1. All NSR algorithms are non-manipulable.
2. Mean-based NR algorithms (such as MW) are manipulable.

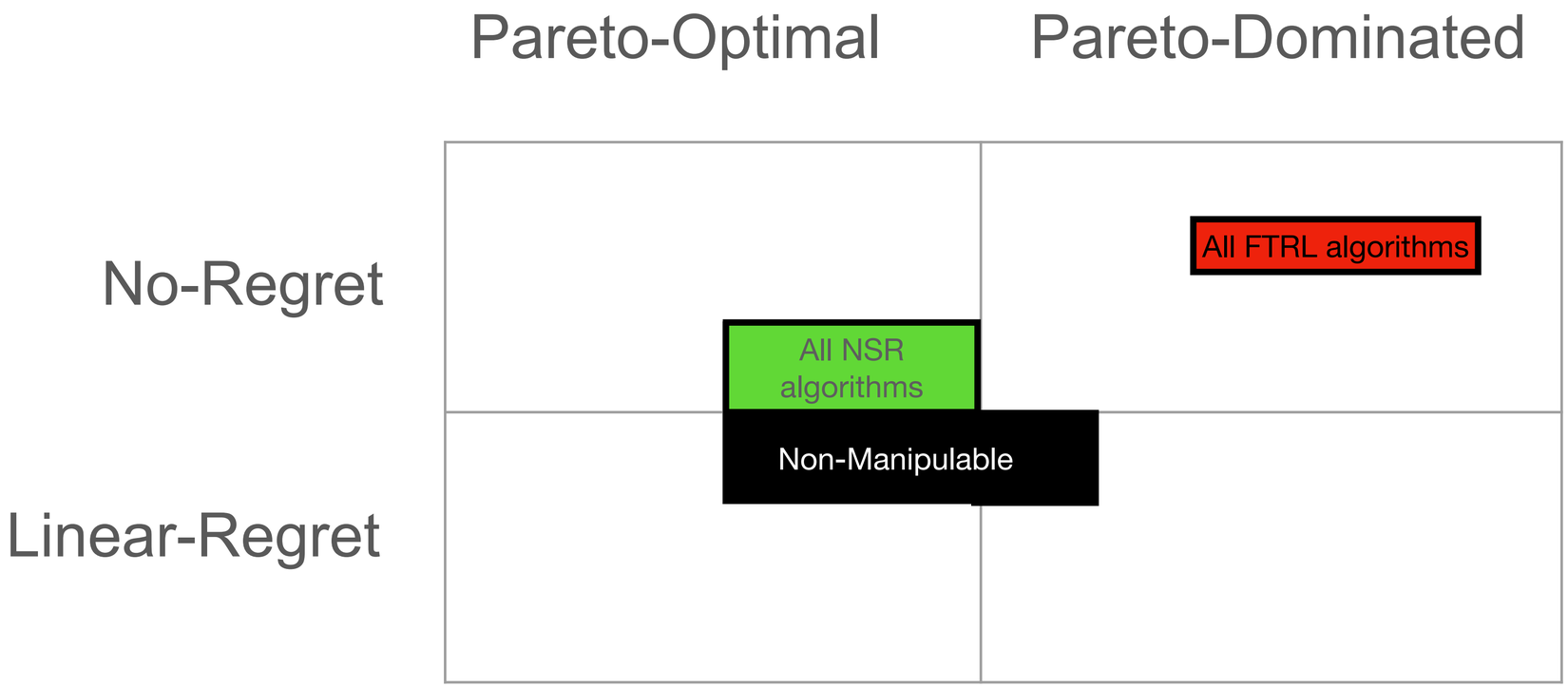
Our Strengthening: Any non-manipulable NR algorithm must be a NSR algorithm.

Useful as a stability property for applications:

For eg: Against a NSR bidder in a repeated auction, setting reserve prices using the Myerson Distribution is optimal!

Ongoing work : Tight characterization of non-manipulability beyond normal-form games

Space of Algorithms



- An infinite continuum of NR + PO as well as NR + Pareto-Dominated
- Unresolved status : Projected Gradient Descent, Regret Matching
- Different results for special subclasses of games, for eg: repeated first-price auctions

Inverse Blackwell Approachability

Minimax Theorems

Von Neumann Minimax Theorem

i.e. No difference between going first and second

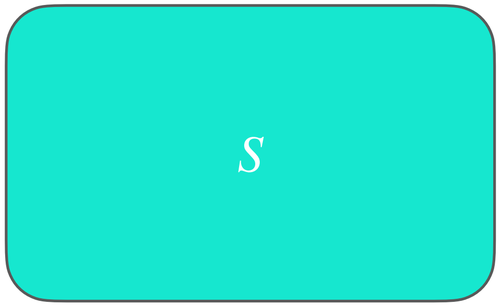
Consider a bilinear function $f : X \times Y \to \mathbb{R}$ for convex action sets X and Y.

$$min_{y \in Y} max_{x \in X} f(x, y) = max_{x \in X} min_{y \in Y}$$

Alternative Statement : Let v = $min_{y \in Y} max_{x \in X} f(x, y)$. There exist an action $x \in X$ such that for all actions $y \in Y$, $f(x, y) \in [v, \infty]$

A Vector Valued Minimax Theorem?

Consider a bilinear vector valued function $f : X \times Y \to \mathbb{R}^k$



A closed, convex set S is said to be response-satisfiable if for all $y \in Y$, there exists $x \in X$ such that $f(x, y) \in S$, i.e. player 1 can guarantee an outcome in S by going second

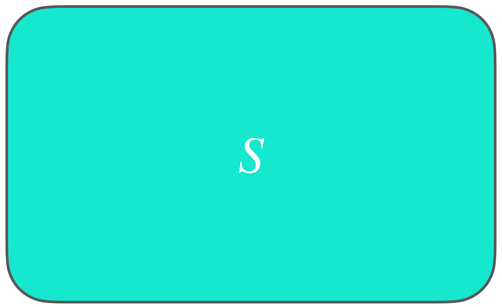
Can Player 1 guarantee an outcome in S by going first instead?

No, But…

Blackwell Approachability Theorem

A Vector valued Minimax Theorem in Algorithm Space

Consider a bilinear vector valued function $f : X \times Y \to \mathbb{R}^k$ for a repeated vector valued game with actions x_t, y_t in the t-th round



A closed, convex set S is said to be response-satisfiable if for all $y \in Y$, there exists $x \in X$ such that $f(x, y) \in S$, i.e. player 1 can guarantee an outcome in S by going second

Theorem (Informal) : There exists an algorithm for player 1 such that for any induced transcript of play, the limit average function value vector approaches S, i.e.,

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} f(x_t, y_t) \in S$$

Almost surely.

- Blackwell 1956 showed the original result and algorithmic construction
- This framework is at the heart of many problems in online learning and multi objective optimization.

The Action Space Game

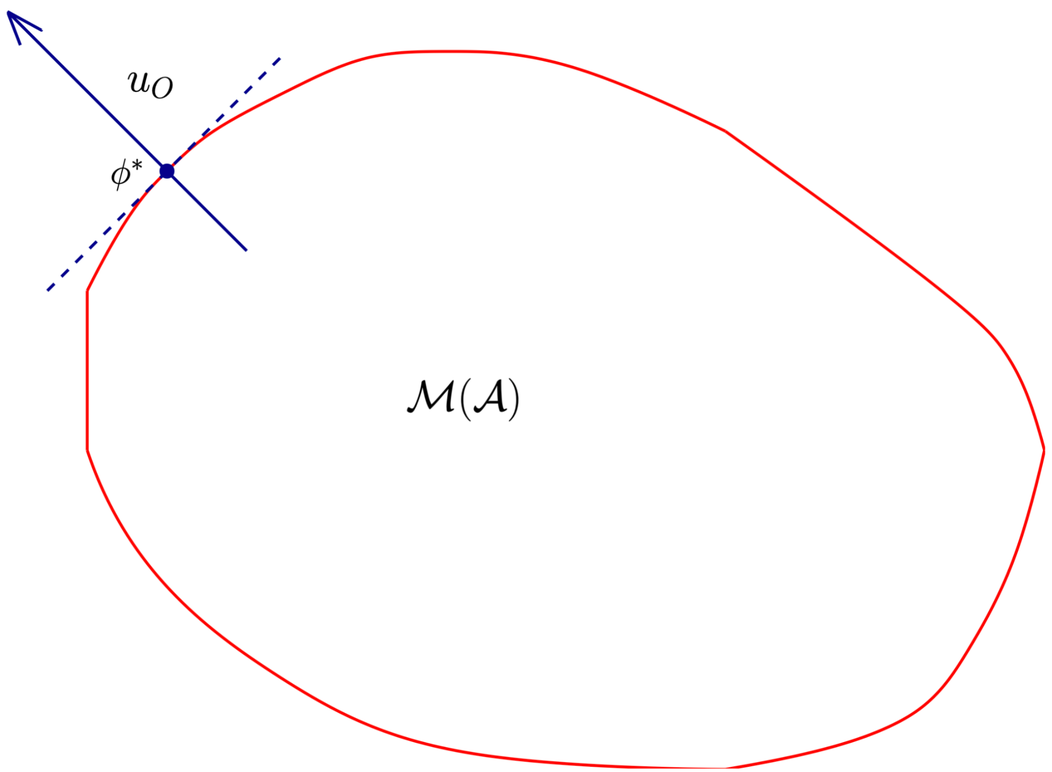
Consider a repeated normal form game with
$X = \Delta^m$ and $Y = \Delta^n$

We define a vector valued function $f : X \times Y \to \mathbb{R}^{mn}$ with
$$f_{ij}(x, y) := x_i y_j$$
i.e. the indicator function for each action pair.

The time average function value $p_T = \frac{1}{T}\sum_{t=1}^{T} x_t \otimes y_t$
tracks the empirical distribution over action pairs. We
call its limit average $\phi = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} x_t \otimes y_t$ the
Correlated Strategy Profile (CSP).

- **Big Idea** : Every learning algorithm A is a Blackwell Approachability algorithms for some set in $\mathbb{R}^{mn}$.
- Define the menu of A to be the inclusion minimal set among all such sets.
- **Claim** : This inclusion minimal set is uniquely defined.

Algorithms -> Menus

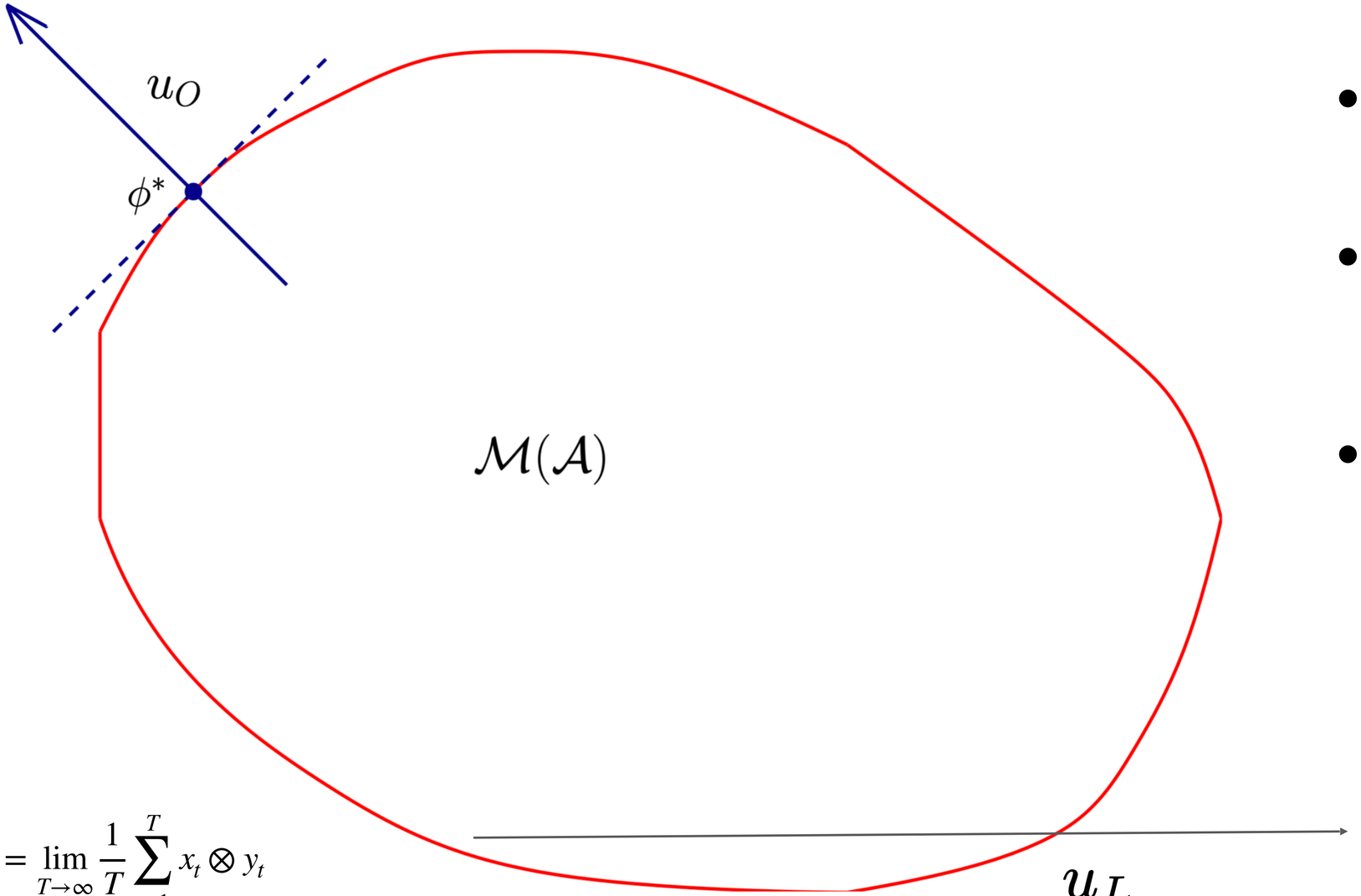


- The convex hull of all CSPs that can be induced by playing against algorithm A
- The inclusion minimal menu for which A is a Blackwell Approachability Algorithm.
- Menus drop some information, but simplify the problem by only preserving outcomes without telling us how to get them.

Each point in the menu is a CSP $\phi = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} x_t \otimes y_t$, that records the empirical statistics of some transcript of play against this algorithm.

Menus are all You Need

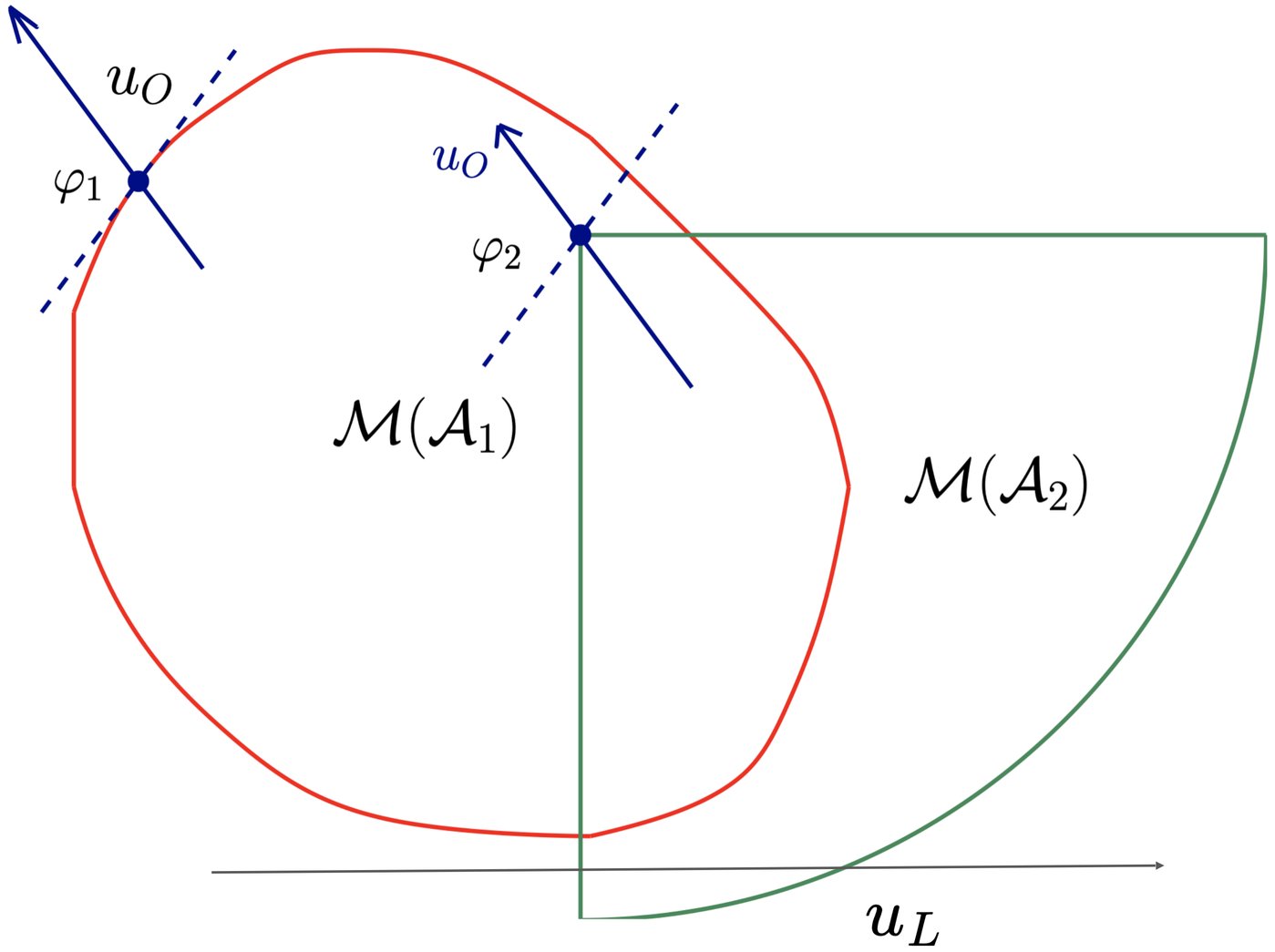
Learner and Optimizer Payoffs



$$\phi = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} x_t \otimes y_t$$

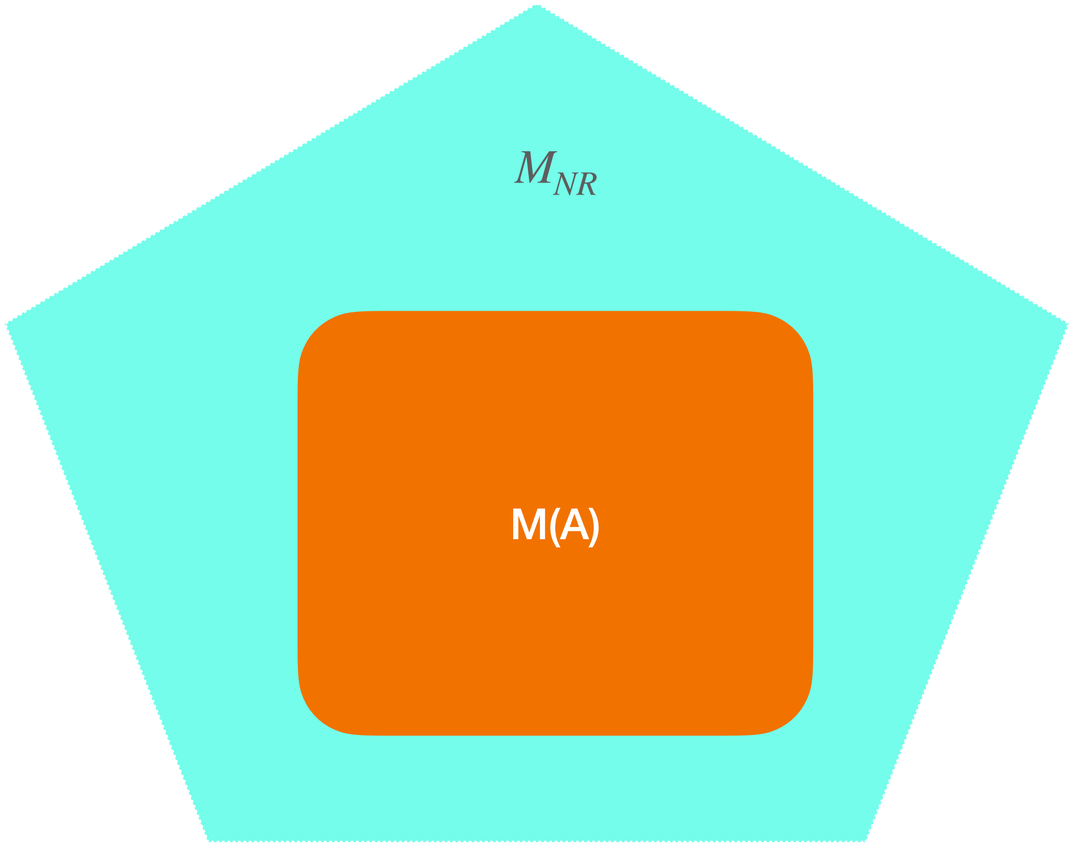
- The optimizer's payoff is a linear objective/ direction in $\mathbb{R}^{mn}$
- The optimizer can simply pick their preferred extreme point maximizing their payoff.
- This, in turn, also fixes the payoff of the learner.

Comparing Two Learning Algorithms



- The optimizer's payoff is a linear objective/ direction in $\mathbb{R}^{mn}$
- The optimizer can simply pick their preferred extreme point maximizing their payoff.
- This, in turn, also fixes the payoff of the learner.
- Simultaneous evaluation with two algorithms and their menus to compare their performance against a given optimizer.

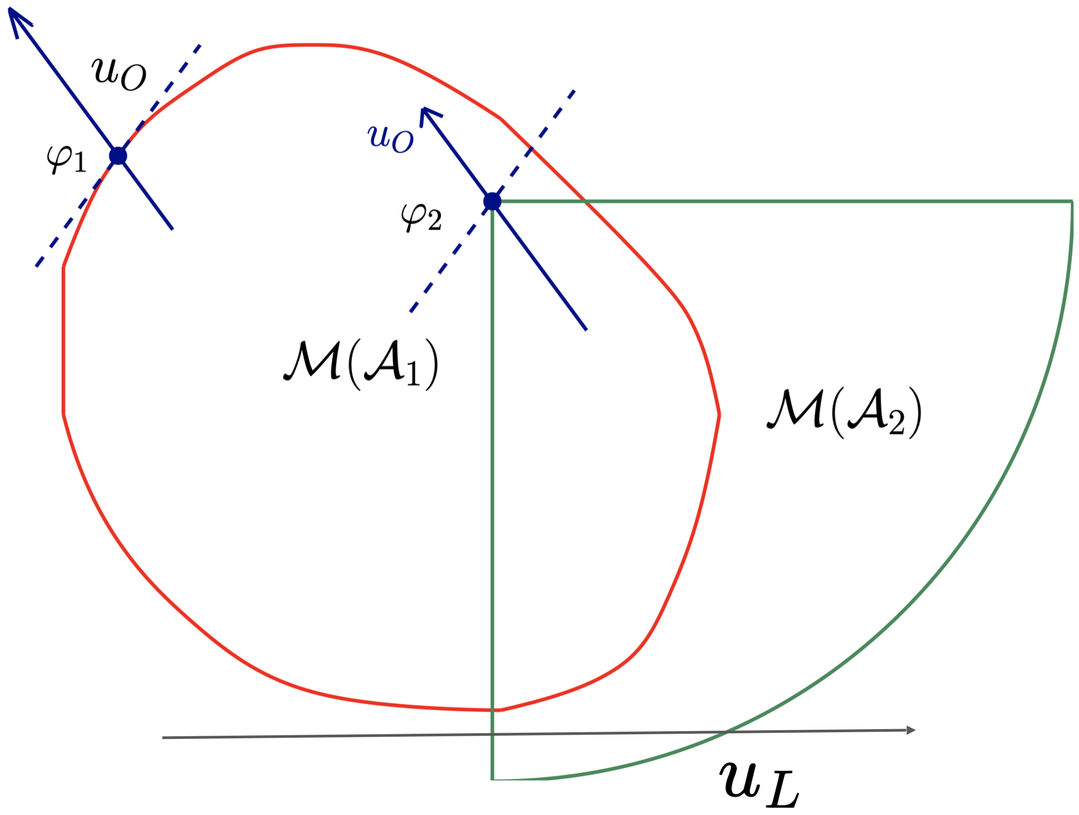
Verifying No-Regret and No-Swap-Regret



$$M_{NR} = \left\{ \phi \in \Delta^{mn} : \sum_{i \in [m], j \in [n]} \phi_{i,j} u_L(i,j) \geq \max_{i^* \in [m]} \sum_{i \in [m], j \in [n]} \phi_{i,j} u_L(i^*,j) \right\}$$

- External Regret and swap-regret are both verifiable using the CSP
  $$\phi = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} x_t \otimes y_t$$
- An algorithm is no-(swap)-regret only if it's menu entirely consists of CSPs with non-positive (swap)-regret.
- Define all CSPs with non-positive regret and call it M_{NR} — this is the one sided coarse correlated equilibrium polytope.
- An algorithm A is no-regret only if $M(A) \subseteq M_{NR}$

Menus are all you Need



- Proving geometric properties about menus translates into proving results about algorithms
- Pareto-Optimality can also be written as a property of menus.
- Non-Manipulability — All extreme points CSPs are product distributions.

For example: this picture is a certificate that algorithm $\mathcal{A}_1$ does not dominate algorithm $\mathcal{A}_2$.

Pareto-Optimality via Menus

Pareto-Optimality via Menus

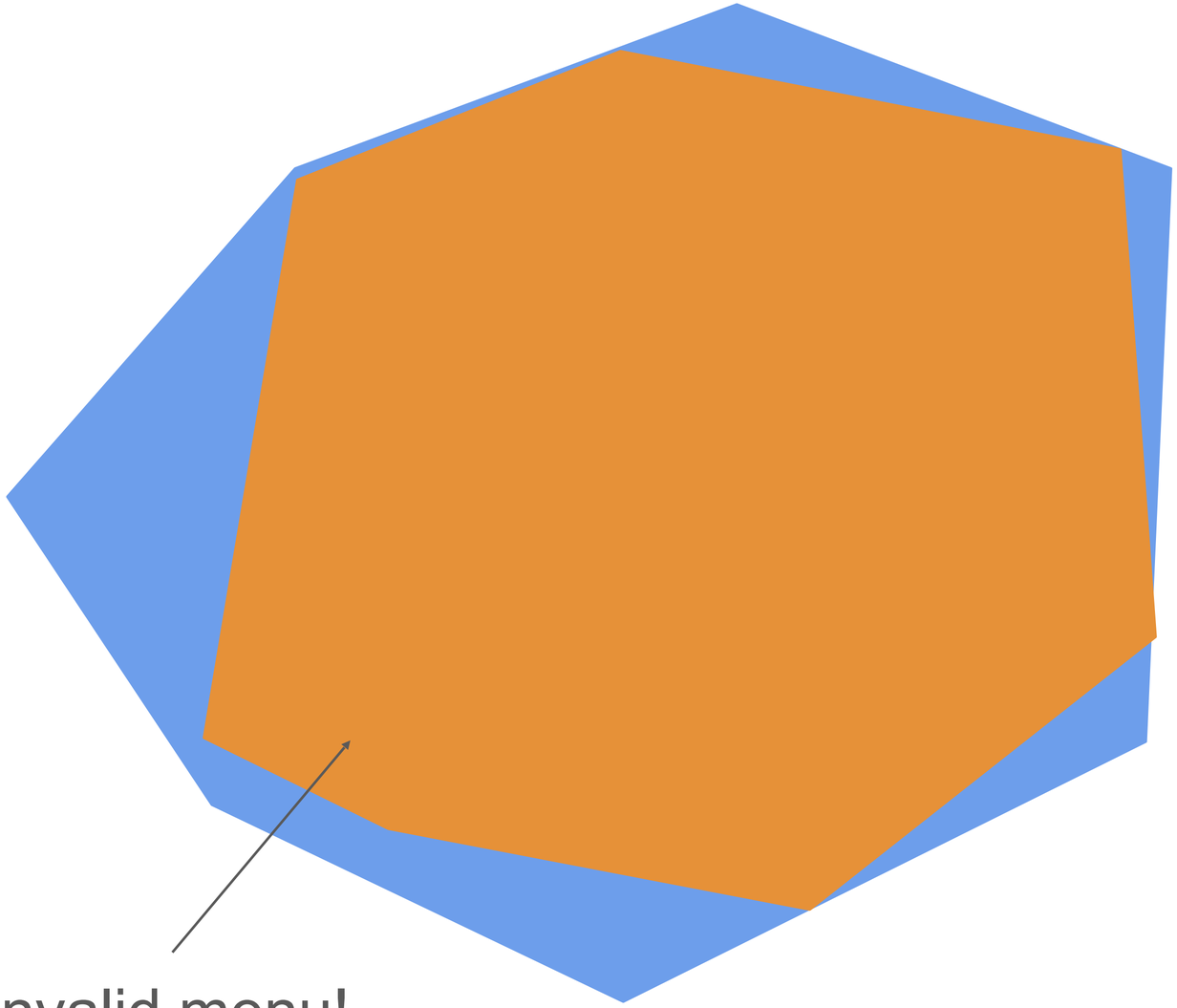
Theorem: All No-Swap Regret Algorithms have the same menu (M_{NSR})

Note : Of particular interest in the light of new, significantly different NSR algorithms ([DDFG 23], [PR 23]) to go with [BM 07]

$\mathcal{M}_{NSR}$

Theorem: All No-Swap Regret Algorithms have the same menu

- Call M_{NSR} the convex hull of all no-swap regret CSPs
- Every no-swap regret menu is tautologically contained within it.
- **Theorem**: M_{NSR} is an inclusion-minimal response-satisfiable set in the action space game.
- **Corollary** : All NSR menus collapse to the same set M_{NSR}.



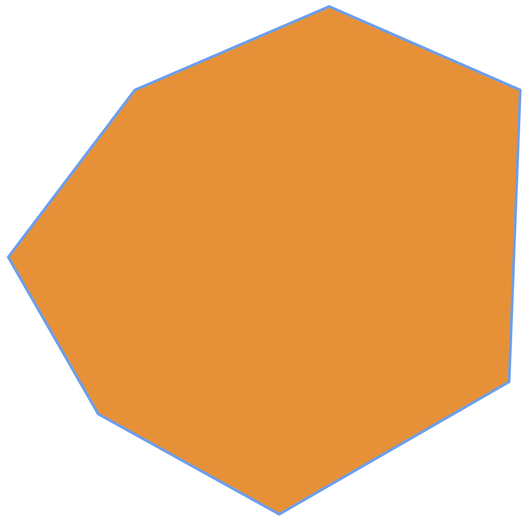
Invalid menu!

$$M_{NSR} = \left\{ \phi \in \Delta^{mn} : \sum_{i \in [m], j \in [n]} \phi_{i,j} u_L(i,j) \geq \max_{\pi:[n]\to[n]} \sum_{i \in [m], j \in [n]} \phi_{i,j} u_L(\pi(i), j) \right\}$$

Inclusion Minimality implies PO!

Theorem: All inclusion minimal menus containing L^+ are PO.

L^+ is the maximum value action pair.



$$M_{NSR} = \left\{ \phi \in \Delta^{mn} : \sum_{i \in [m], j \in [n]} \phi_{i,j} u_L(i,j) \geq \max_{\pi:[n]\to[n]} \sum_{i \in [m], j \in [n]} \phi_{i,j} u_L(\pi(i), j) \right\}$$

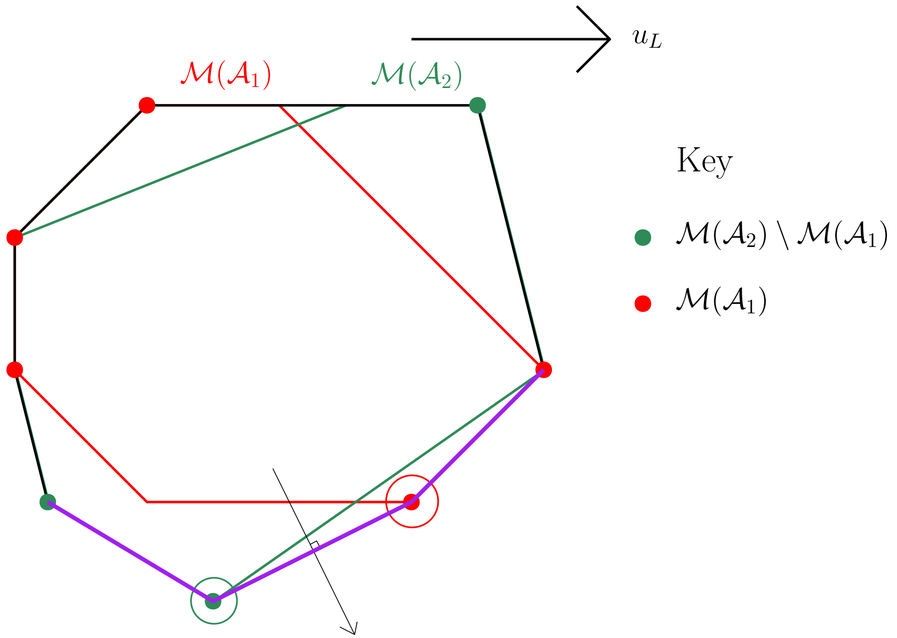
- All NSR algorithms have the same menu.
- This menu is minimal, and therefore Pareto-Optimal.

- M_{NSR} is a polytope whose extreme points are Stackelberg equilibria of the one-shot game with the optimizer as the leader and the learner as a follower (recovering a result of DSS 19).
- Thus, all NSR algorithms are non-manipulable.
- Playing a No-swap-regret algorithm is a way for a learner to exchange the power of commitment with the optimizer, an idea we have explored in the context of algorithmic collusion.

Inclusion Minimality and PO

Lemma : If M_1 contains L^+ and $M_2 \backslash M_1 \neq \emptyset$, then there is an Optimizer payoff u_O such that

$$u_L(M_1, u_O) > u_L(M_2, u_O)$$



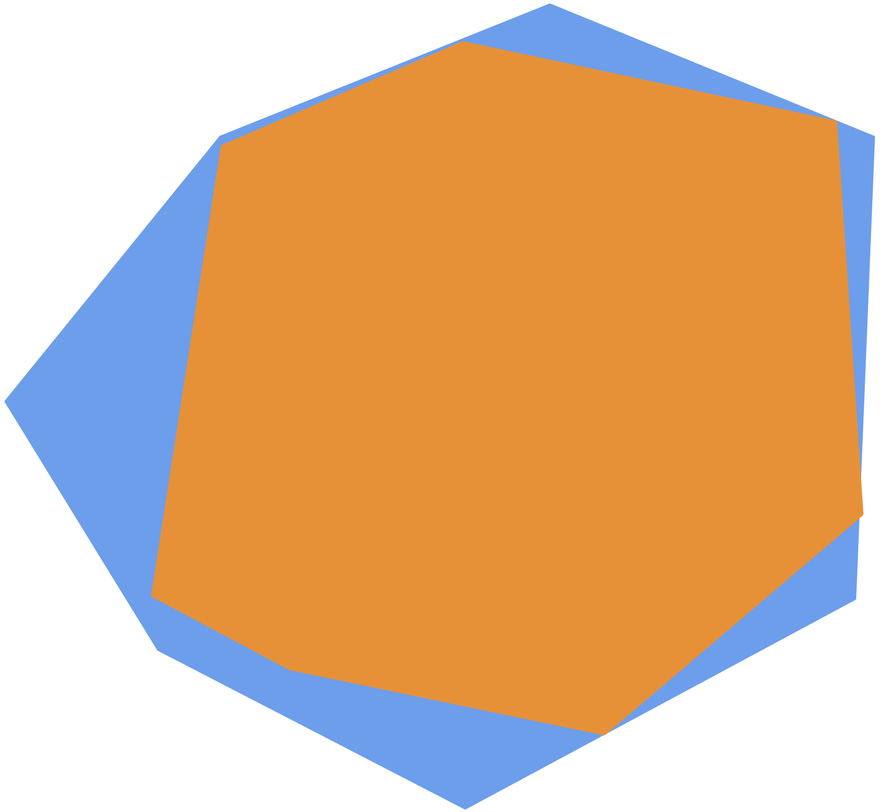
Key Idea : To show a menu is Pareto-optimal, only need a single certificate of non-domination against any other menu.

Toy Proof : Where both menus are polytopes, a path following argument suffices.

1. Start with an “extra” vertex in M_2
2. Construct a path of strictly increasing u_L value
3. Find a “crossover” edge

FTRL is Pareto-dominated

Pareto-Domination



If Algorithm A is no-regret, then $M_{NSR} \subseteq M(A)$

- A route to Pareto-Domination: Show that certain “bad” points can be removed from the menu.
- Tricky Aspect : While menus are upwards closed, removing points might break their response-satisfiable property.
- Silver Lining : Every no-regret menu contains the canonical no-swap-regret menu.

FTRL

Only moves within $o(T)$ of being the historical best-response action get non-trivial, i.e., $\Omega_T(1)$ mass.

Given that R is continuous and strongly-convex, and $\eta_T = \frac{1}{o(T)}$:

$$y_t = \arg\max_{y \in \Delta^n} \left(\sum_{s=1}^{t-1} u_L(x_s, y) - \frac{R(y)}{\eta_T} \right)$$

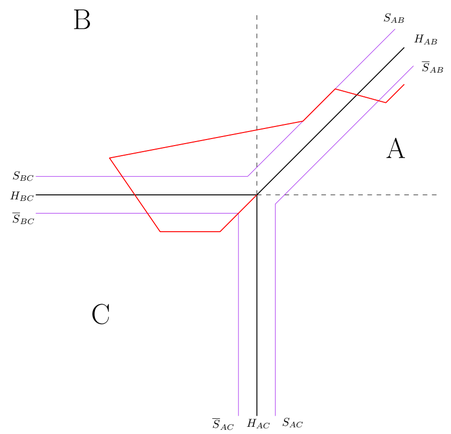
All Follow-the-Regularized Leader type algorithms, including Multiplicative Weights (Hedge), Online Gradient Descent are Mean-Based No-Regret Algorithms

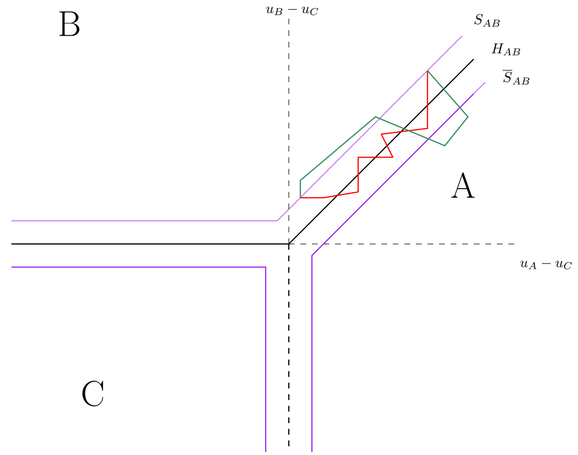
- FTRL algorithms can be shown to contain "bad" points via the "mean-based" property.
- Deleting bad points via showing that FTRL algorithms have a polytope menu.

FTRL Menus are Polytopes

Mean-Based Trajectories

Trajectory has a ``clear" leader for all but o(T) time steps.





- FTRL algorithms have a special state space.
- Key Idea: Show that w.l.o.g, the optimal response trajectory (through the FTRL state space) satisfies some constraints.
- These constraints ensure that the menu has a finite number of extreme points, i.e., is a polytope.

Connection: All FTRL algorithms are mean-based, i.e. they almost always play ``clear" leaders.

Menus: A geometric view of algorithms

- **Algorithms -> Menus**
  - Useful for analysis of pre-existing algorithms, capture various properties of interest

- **Menus -> Algorithms**
  - Fully understand what convex sets are “feasible” menus
  - Useful new tool for algorithm design: design the space of possibilities you want (as long as it’s response-satisfiable), then “invert” the menu to obtain an algorithm —> via Blackwell + A Padding Argument

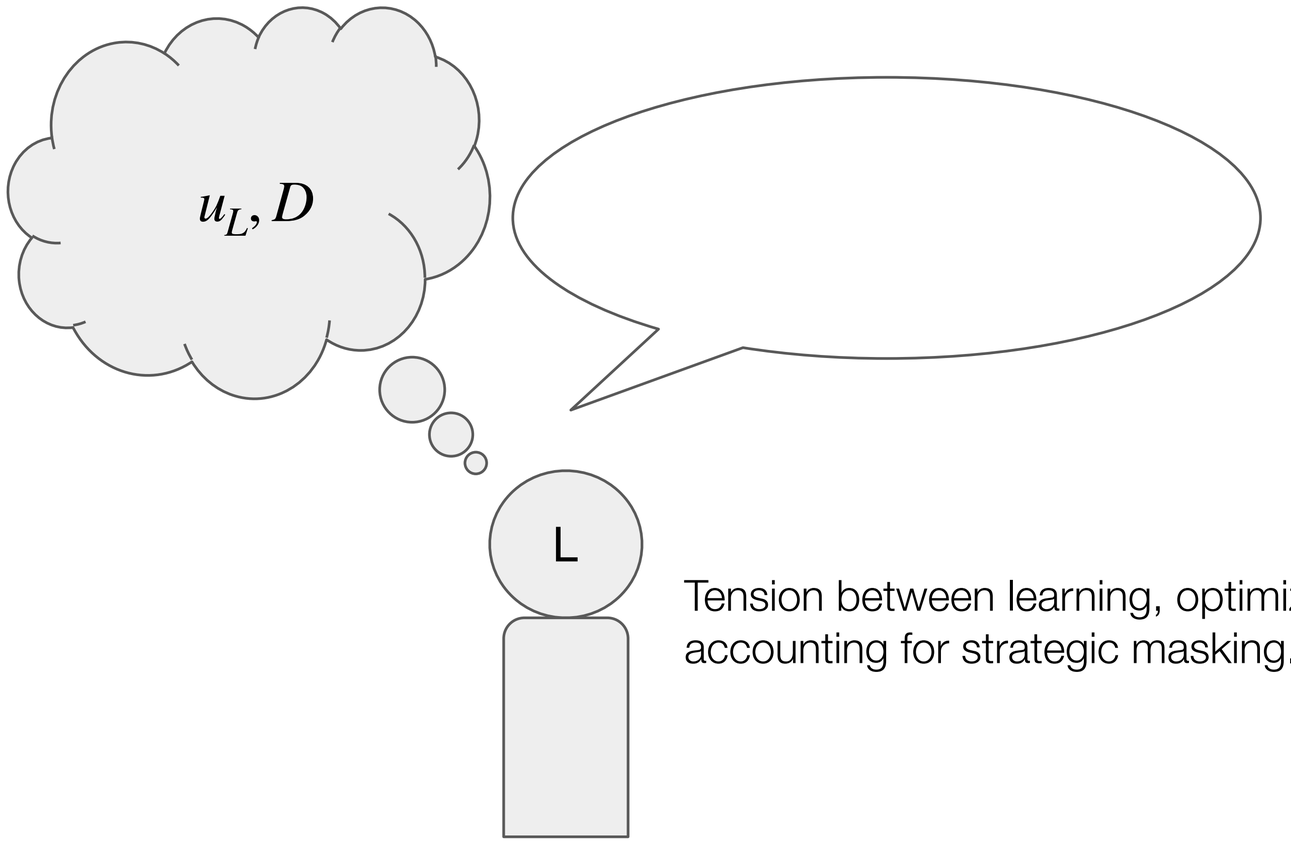
Learning to Play Against Unknown Opponents

Eshwar Ram Arunachaleswaran, Natalie Collina, Jon Schneider

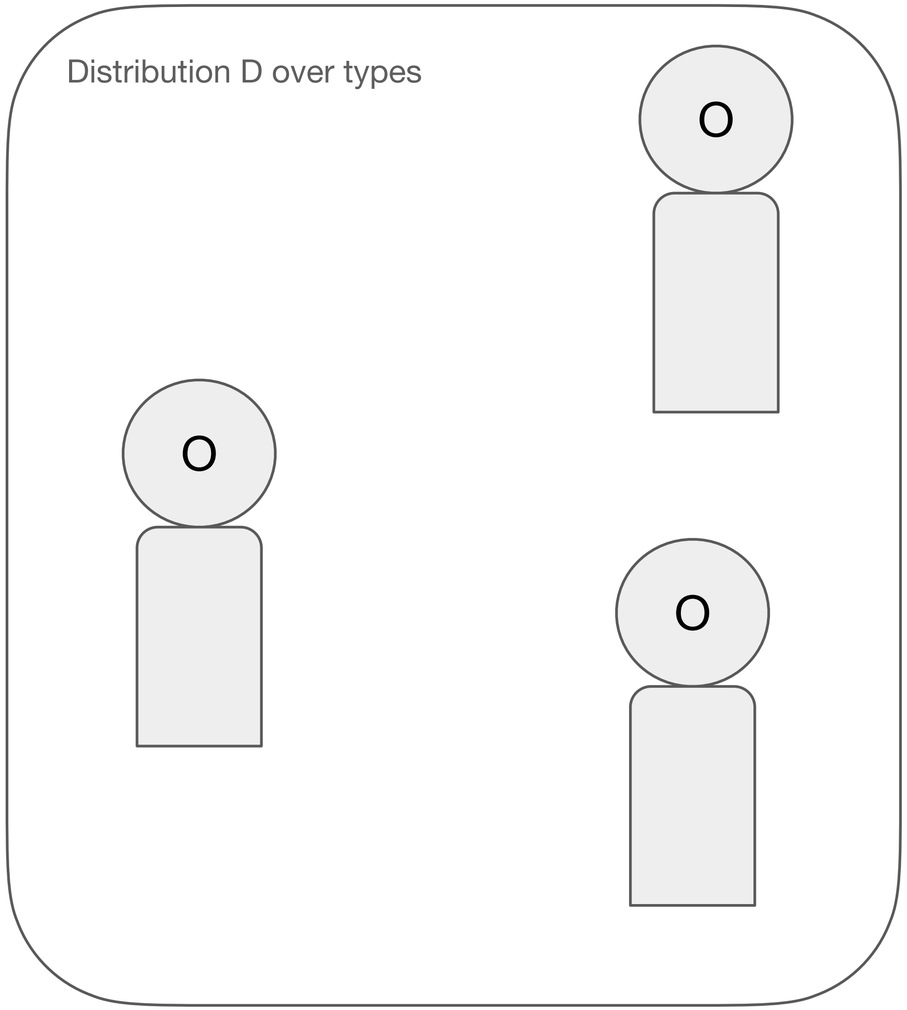
In Submission (Arxiv soon)

Public Prior over Optimizer Types

What is the optimal (No-Regret) algorithm to commit to?

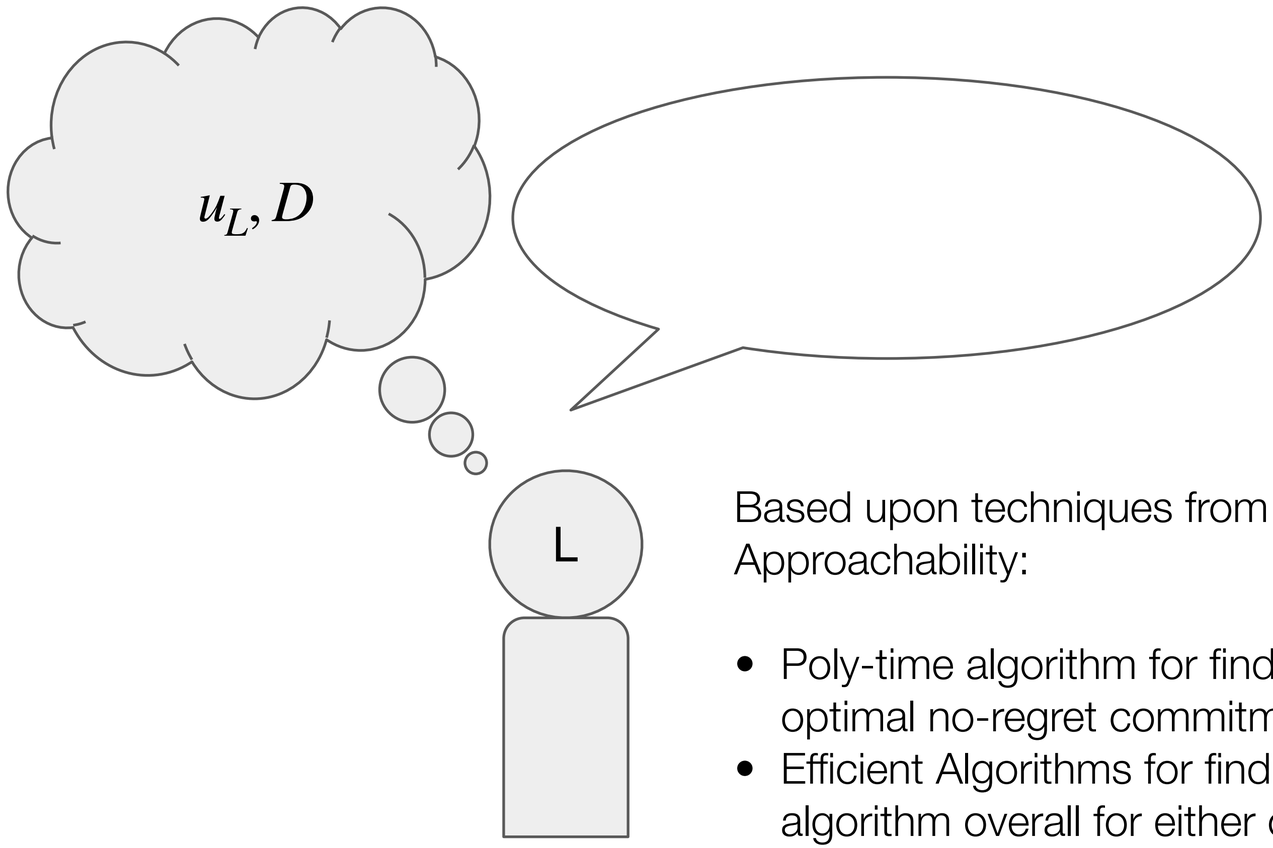


Tension between learning, optimization and accounting for strategic masking.



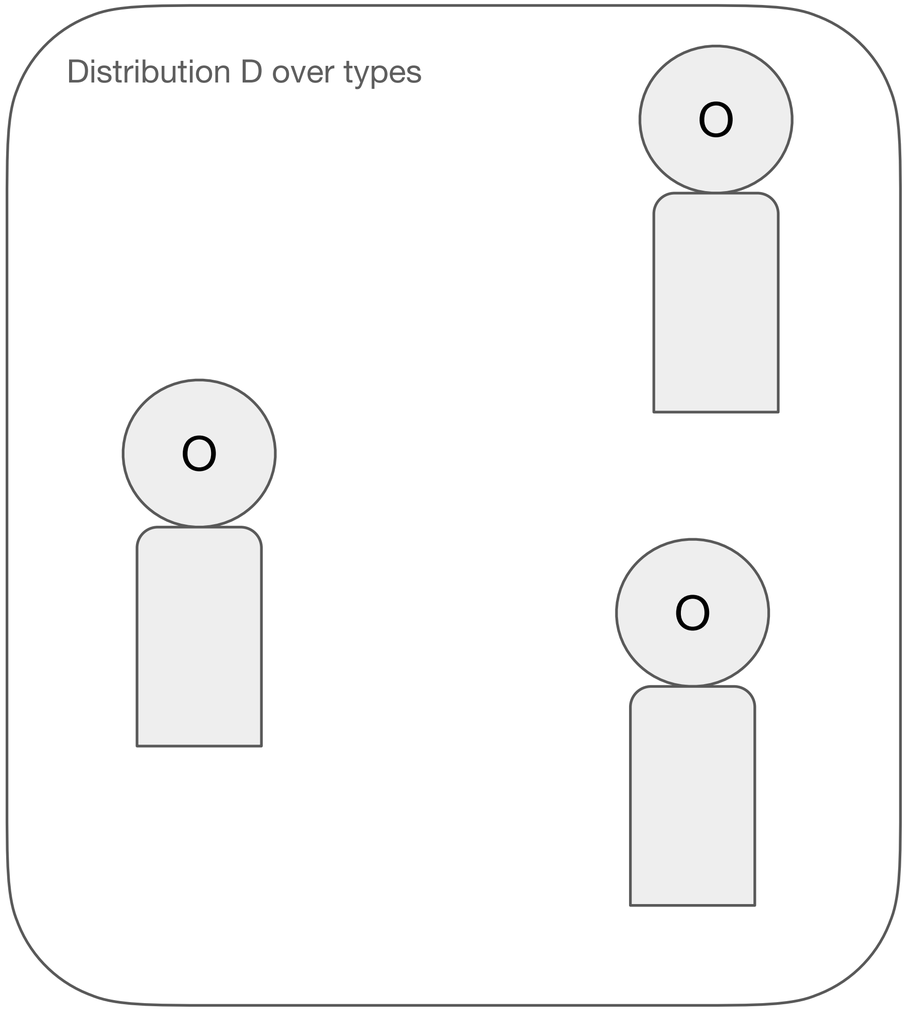
Public Prior over Optimizer Types

What is the optimal (No-Regret) algorithm to commit to?



Based upon techniques from Blackwell Approachability:

- Poly-time algorithm for finding the optimal no-regret commitment
- Efficient Algorithms for finding optimal algorithm overall for either constant sized games or constant number of types.



About me

Hi, I'm Eshwar Ram Arunachaleswaran

- Final Year PhD Student, University of Pennsylvania (Currently visiting the Simons Institute as a participant in the year long program on large language models and transformers)
- Advisors : Sampath Kannan, Anindya De
- Research Interests: Algorithmic Game Theory, Online Learning
- Job Search : Postdocs and Industry Research Positions
- Current Problem Areas : Learning Algorithms for Games, (Multi)Calibration, Omniprediction, Generalized Stackelberg Equilibria, Emergent Coordination/Algorithmic Collusion

Algorithmic Fairness and Machine Learning

- Oracle Efficient Algorithms for Groupwise Regret (ICLR 2024): Proposes regret-based approaches for fairness at group levels.
- Wealth Dynamics Over Generations: Analysis and Interventions (SATML 2023): Analyzes intergenerational dynamics and interventions for fairness.
- Pipeline Interventions (ITCS 2021, Math. Operations Research 2022): Addresses fairness in biased decision-making pipelines.

Online Learning and Repeated Games

- Algorithmic Collusion Without Threats (ITCS 2025): Models emergent collusion mechanisms in repeated games.
- An Elementary Predictor Obtaining Distance to Calibration (SODA 2025)
- Pareto-Optimal Algorithms for Learning in Games (EC 2024)
- Efficient Stackelberg Strategies for Finitely Repeated Games (AAMAS 2023)
- Learning to Play Against Unknown Opponents (in submission)

Fair Division and Equilibrium Computation

- Fully Polynomial-Time Approximation Schemes for Fair Rent Division (SODA 2019, Math. Operations Research 2022):
- Fair and Efficient Cake Division with Connected Pieces (WINE 2019):
- Fair Division with a Secretive Agent (AAAI 2019)

Thanks