Certificates in Extensive-Form Games

Brian Hu Zhang and Tuomas Sandholm, “Small Nash Equilibrium Certificates in Very Large Games”, NeurIPS-20:
https://arxiv.org/abs/2006.16387

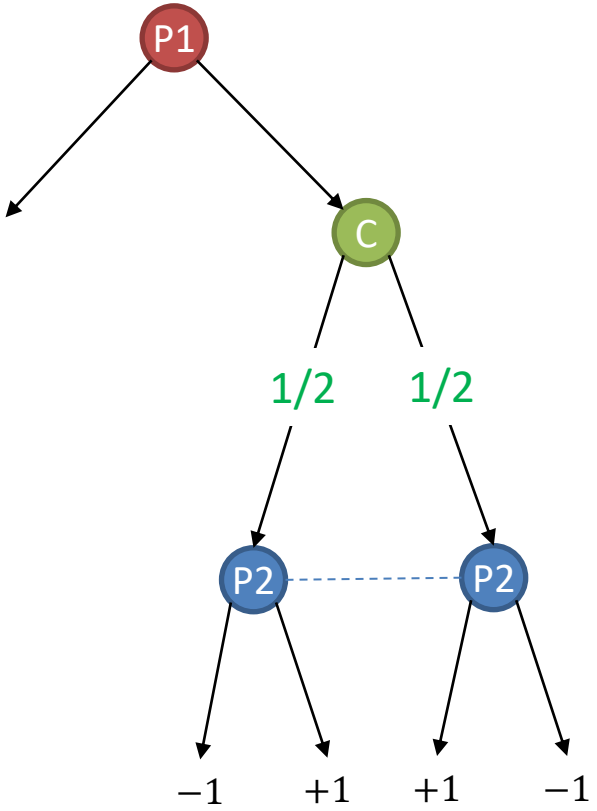
Brian Hu Zhang and Tuomas Sandholm, “Finding and Certifying (Near-)Optimal Strategies in Black-Box Extensive-Form Games”, AAAI-21:
https://arxiv.org/abs/2009.07384

Learning to Play Black-Box Games

- **Deep Reinforcement Learning** (*AlphaZero* [Silver et al, 2017], *AlphaStar* [Vinyals et al, 2019], *OpenAI Five* [Berner et al, 2019], etc)
  - Good practical performance
  - **Issue:** No exploitability bounds
- **Bandit Regret Minimization** [Farina et al, 2020]
  - Converges to ε-equilibrium after $\mathrm{poly}(N, 1/\varepsilon)$ game samples (N = number of nodes in game tree)
  - **Issue:** Worst-case exploitability bounds are trivial until number of iterations is much larger than N, need to expand rest of game tree to compute *ex-post* exploitability guarantee
- **Certificates** [**This work**]
  - Compute Nash equilibrium by incrementally expanding game tree
  - Exploitability bounds always computable *ex post* without expanding remainder of tree!

Pseudogames and Certificates

Pseudogame: Game without known utilities on all terminal nodes



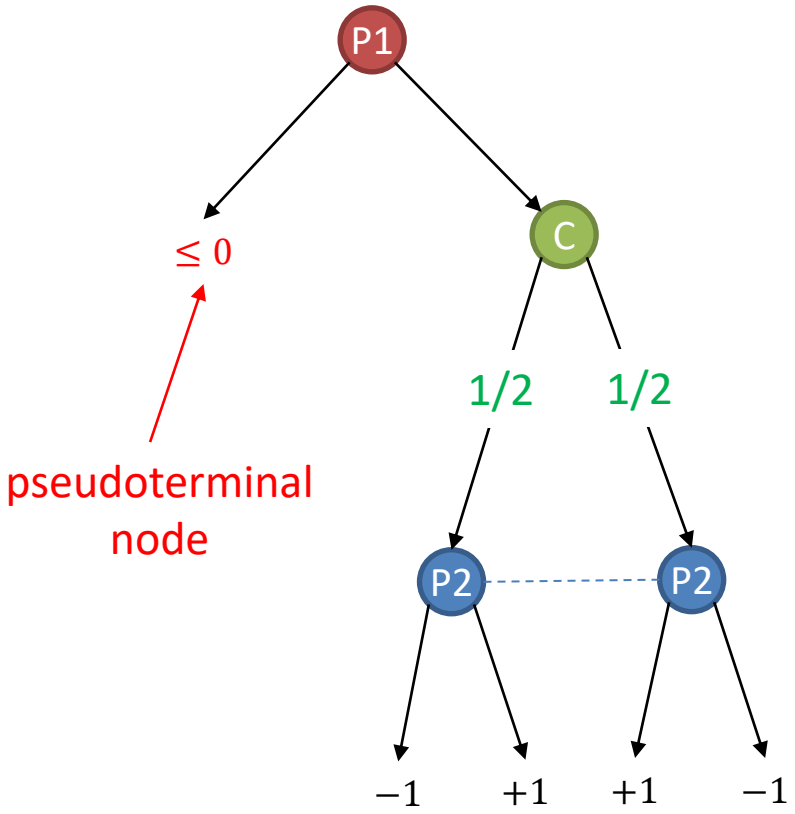
Pseudogames and Certificates

Pseudogame: Game without known utilities on all terminal nodes

Think: partially-expanded game tree, “alpha-beta” style

In zero-sum land, gives rise to **two games**:

- a *lower-bound game* in which rewards are optimistic for P2, and
- an *upper-bound game* in which rewards are optimistic for P1

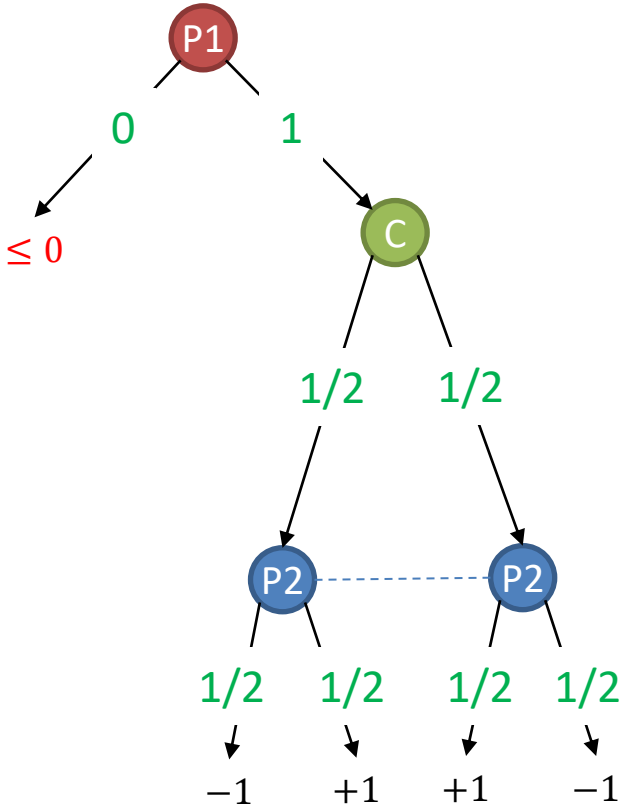


Pseudogames and Certificates

(Approximate) Nash equilibrium in a pseudogame: strategy profile in which every player is *provably* playing an (approximate) best response (irrespective of what happens at pseudoterminal nodes)

Results in Nash equilibrium regardless of what the pseudoterminal node hides!

(Approximate) Certificate: Pseudogame created from partial expansion of a full game + (approximate) Nash equilibrium of that pseudogame



Small Certificates

- **Question:** When do small ε-certificates (size $O\left(N^c \mathrm{poly}(1/\varepsilon)\right)$ for some $c < 1$, where N is the number of nodes) exist?

When do Small Certificates Exist?

- **Answer #1:** They exist in **perfect-information zero-sum games with no nature randomness**,

…under reasonable assumptions about the game tree (uniform branching factor and depth, alternating moves)

  - **Proof:** The optimal alpha-beta search tree is a certificate of size $\approx \sqrt{N}$**.**

Small Certificates

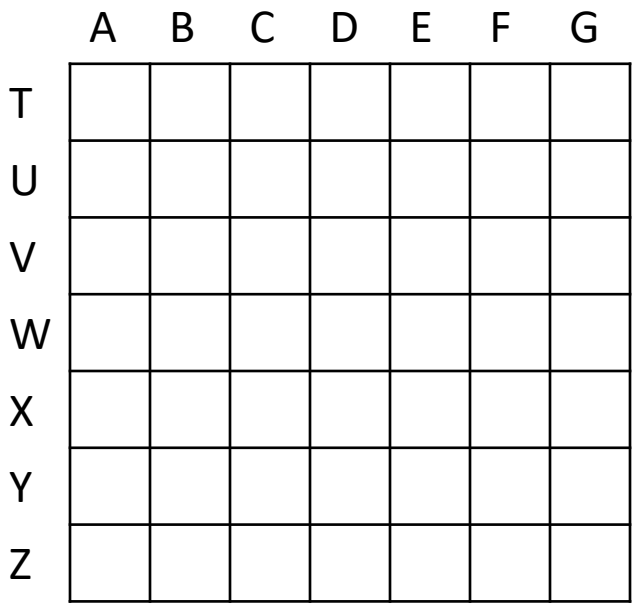
Answer #2: They exist in (squarish) **normal-form games.**

Proof:

Consider an $m \times m$ normal-form game.

Lipton et al, 2003:

ε-Nash equilibrium exists where each player mixes between $\log(m)\ /\varepsilon^2$ pure strategies.



Small Certificates

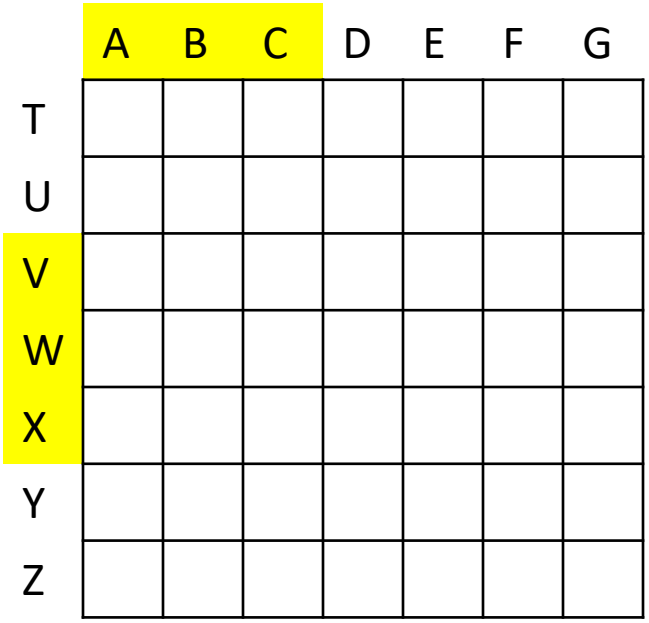
Answer #2: They exist in (squarish) **normal-form games.**

Proof:

Consider an $m \times m$ normal-form game.

Lipton et al, 2003:

ε-Nash equilibrium exists where each player mixes between $\log(m)\ /\varepsilon^2$ pure strategies.



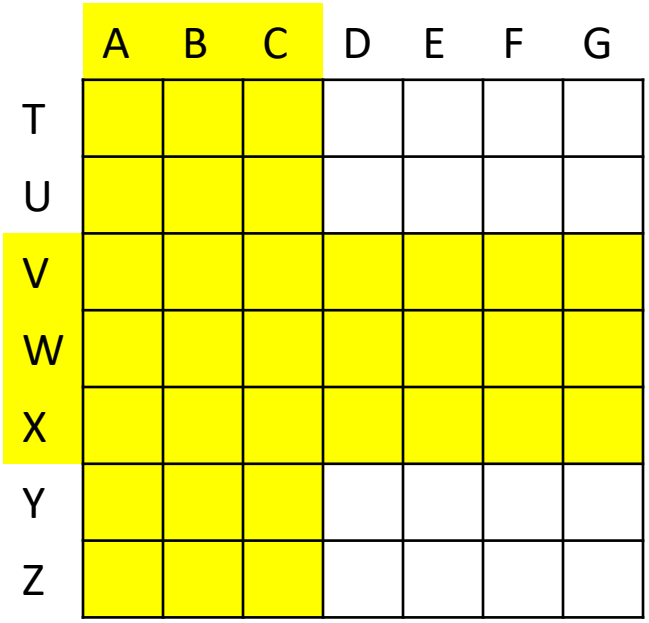
Small Certificates

Answer #2: They exist in (squarish) **normal-form games.**

Proof:

We only need those rows and columns!

$\Rightarrow O(m \log(m)/\varepsilon^2)$-size certificate



Small Certificates

So, small certificates exist in games where the players have **perfect information** or **no information**.

What about in between?

Unfortunately, no. ☹

Bad News

Counterexample: Consider this game:

- Matching pennies
- repeated k times, each round worth $1/k$ points.
- After each round, both players learn what the other played

Game tree size: 4^k

Theorem: Any ε-certificate of this game must at least $4^{k(1-2\varepsilon)}$ nodes.

Bad News

Theorem: Any ε-certificate of k-repeated matching pennies must at least $4^{k(1-2\varepsilon)}$ nodes

Proof: We will show:

Lemma: Consider a certificate of k-matching pennies with C terminal nodes. Then, P2 has an optimistic best response in which she loses no more than $\log_{16} C$ rounds

This would be enough, because then we would need $\log_{16} C \geq k\left(\frac{1}{2} - \varepsilon\right)$, or $C \geq 4^{k(1-2\varepsilon)}$

Bad News

Lemma: Consider a certificate of k-matching pennies with C terminal nodes. Then, P1 has an optimistic best response in which she loses no more than $\log_{16} C$ rounds

Proof: Whiteboard

More Bad News

Theorem: It is NP-hard to approximate the smallest certificate of an extensive-form zero-sum game, to better than an $O(\log N)$ multiplicative factor.

Proof Idea: Reduction from set cover.

Oracle Model

Assume access to an **oracle** that allows us to query any node h to obtain:

- upper and lower bounds on the future utility after h
- the player to act at h, if any, and that player's information
- if the player to act is nature, the exact nature distribution

Goal:

- *Compute and verify "ex-post"* approximate equilibria *with only black-box access*
- Output both an equilibrium strategy **and** a bound ε on exploitability

More Bad News

Theorem: With only an oracle for an extensive-form zero-sum game, there is no equilibrium-finding algorithm that runs in time polynomial in the size of the smallest certificate.

Proof: One-player "SAT" games: certificate of size $O(\log N)$ exists, but clearly no sublinear-time algorithm.

Let's Try Anyway

Repeat until satisfied:

- **Solve** both the upper- and lower-bound pseudogames *exactly* (with e.g., an LP solver)
- **Create** the next pseudogame, by expanding all pseudoterminal nodes in the support of the **optimistic profile** (in which the max-player her equilibrium strategy in the upper-bound game, and the min-player plays her strategy in the lower-bound game)

Output: Pessimistic profile, and $\varepsilon =$ difference in values between upper- and lower-bound pseudogames

Intuition: In the perfect-information setting with no nature randomness, **it's just alpha-beta search**

Let's Try Anyway

Repeat until satisfied:

- **Solve** both the upper- and lower-bound pseudogames *exactly* (with e.g., an LP solver)
- **Create** the next pseudogame, by expanding all pseudoterminal nodes in the support of the **optimistic profile** (in which the max-player her equilibrium strategy in the upper-bound game, and the min-player plays her strategy in the lower-bound game)

Output: Pessimistic profile, and $\varepsilon =$ difference in values between upper- and lower-bound pseudogames

Theorem (Correctness): If the pessimistic profile is not a Nash equilibrium, then the second step expands at least one node.

Let's Try Anyway

Repeat until satisfied:

- **Solve** both the upper- and lower-bound pseudogames *exactly* (with e.g., an LP solver)
- **Create** the next pseudogame, by expanding all pseudoterminal nodes in the support of the **optimistic profile** (in which the max-player her equilibrium strategy in the upper-bound game, and the min-player plays her strategy in the lower-bound game)

Output: Pessimistic profile, and $\varepsilon =$ difference in values between upper- and lower-bound pseudogames

"Works" even on games that have infinitely large trees or infinite/unbounded rewards!

Experiments

game	size of game		size of certificate			
	nodes	infosets	nodes		infosets	
search game	234,705	11,890	13,682	5.8%	532	4.5%
4-rank PI Goofspiel	2,229	1,653	275	12.3%	110	6.7%
5-rank PI Goofspiel	55,731	41,331	2,593	4.7%	957	2.3%
6-rank PI Goofspiel	2,006,323	1,487,923	21,948	1.1%	7,584	0.5%
4-rank Goofspiel	2,229	738	614	27.5%	117	15.9%
5-rank Goofspiel	55,731	9,948	11,415	20.5%	2,160	21.7%
6-rank Goofspiel	2,006,323	166,002	266,756	13.3%	15,776	9.5%
3-rank random Goofspiel	1,066	426	309	29.0%	92	21.6%
4-rank random Goofspiel	68,245	17,432	16,416	24.1%	3,270	18.8%
5-rank random Goofspiel	8,530,656	1,175,330	1,854,858	21.7%	241,985	20.6%
5-rank Leduc	∞	∞	26,306	—	2,406	—
9-rank Leduc	∞	∞	137,662	—	6,811	—
13-rank Leduc	∞	∞	337,312	—	12,171	—

Simulators

Assume access to a **simulator:**

- Allows us to play through the game **from the perspective of all players at once**
- Gives player to act, acting player's information, bounds on future utility, and valid actions
- **Does not** give nature distribution; only gives a single sample
- **Does not** allow saving and rewinding. Must perform complete play-throughs

Goal:

- *Compute and verify "ex-post"* approximate equilibria *with only black-box access*
- Output both an equilibrium strategy **and** a bound ε on exploitability
- **Want**: correctness with high probability, say, $1 - T^{-\gamma}$ for some $\gamma > 0$ after T iterations.

Lower Bounds

Theorem: Consider any algorithm with the following guarantee: For some constant $\gamma > 0$, given a zero-sum game in our black-box setting, with T game samples, the algorithm outputs a pair of strategies (x, y) **and a bound** ε_T such that, with probability $1 - O(T^{-\gamma})$, (x, y) is an ε_T-Nash equilibrium. Then

$$\varepsilon_T = \Omega\left(\sqrt{\frac{\log T}{T}}\right).$$

Our goal: Match this bound.

Main Tool: Pseudogames as Confidence Bounds

- At **nodes that have not yet been expanded**, use bounds given by simulator
- At **nature nodes** h, give each player reward bounded by $[-\rho, \rho]$, where

$$\rho = \Delta_h \sqrt{\frac{1}{2t_h} \log \frac{1}{\delta}}$$

range of utilities possible from h

times h has been reached

confidence parameter

Main Tool: Pseudogames as Confidence Bounds

- At **nodes that have not yet been expanded**, use bounds given by simulator.
- At **nature nodes** h, give each player reward bounded by $[-\rho, \rho]$, where

$$\rho = \Delta_h \sqrt{\frac{1}{2t_h} \log \frac{1}{\delta}}$$

Intuition: ρ represents the **uncertainty** in the nature distribution at h

Main Tool: Pseudogames as Confidence Bounds

- At **nodes that have not yet been expanded**, use bounds given by simulator.
- At **nature nodes** h, give each player reward bounded by $[-\rho, \rho]$, where

$$\rho = \Delta_h \sqrt{\frac{1}{2t_h} \log \frac{1}{\delta}}$$

Intuition: It looks like UCB. That is not a coincidence.

Choice of Confidence Bound

During equilibrium computation, values of children are changing, so we need to use a Hoeffding bound to be robust:

$$\rho = \Delta_h \sqrt{\frac{1}{2t_h} \log \frac{1}{\delta}}$$

During best response computation, strategy profiles after h are fixed by induction, so we can use a tighter empirical Bernstein bound [Maurer & Pontil '09]:

$$\rho = S \sqrt{\frac{2}{t_h} \log \frac{2}{\delta}} + \frac{7\Delta'_h}{3(t_h - 1)} \log \frac{2}{\delta}$$

where S is the unbiased sample standard deviation, and Δ'_h is the range of possible utilities from h **under the fixed strategy profile**, which may be much smaller than Δ_h

Main Tool: Pseudogames as Confidence Bounds

Theorem: For appropriate choice of $\delta = \mathrm{poly}\left(\frac{1}{T}, N\right)$, with high probability, at every time, for every strategy profile, for every player the true reward of the player is bounded by the pessimistic and optimistic rewards achieved in the confidence bound pseudogame.

("Confidence bounds are actually bounds")

Zero-Sum LP-Based Algorithm

Repeat T times:

- **Solve** both the upper- and lower-bound pseudogames *exactly* (with e.g., an LP solver)
- **Sample** one play-through from the optimistic profile (in which the max-player her equilibrium strategy in the upper-bound game, and the min-player plays her strategy in the lower-bound game)
- **Create** the next pseudogame:
  - **Expand** all encountered nodes
  - **Update** empirical nature distributions of nature nodes sampled during play

Output: Pessimistic profile, and $\varepsilon_T =$ difference in values between upper- and lower-bound pseudogames

Intuition: In the perfect-information setting with no nature randomness, **it's just alpha-beta search**

Zero-Sum LP-Based Algorithm

Repeat T times:

- **Solve** both the upper- and lower-bound pseudogames *exactly* (with e.g., an LP solver)
- **Sample** one play-through from the optimistic profile (in which the max-player her equilibrium strategy in the upper-bound game, and the min-player plays her strategy in the lower-bound game)
- **Create** the next pseudogame:
  - **Expand** all encountered nodes
  - **Update** empirical nature distributions of nature nodes sampled during play

Output: Pessimistic profile, and ε_T = difference in values between upper- and lower-bound pseudogames

Intuition: In the one-player "multi-armed bandit" setting, **it's UCB (up to a constant factor).**

Zero-Sum LP-Based Algorithm

Repeat T times:

- **Solve** both the upper- and lower-bound pseudogames *exactly* (with e.g., an LP solver)
- **Sample** one play-through from the optimistic profile (in which the max-player her equilibrium strategy in the upper-bound game, and the min-player plays her strategy in the lower-bound game)
- **Create** the next pseudogame:
  - **Expand** all encountered nodes
  - **Update** empirical nature distributions of nature nodes sampled during play

Output: Pessimistic profile, and ε_T = difference in values between upper- and lower-bound pseudogames

Advantage: Sample-efficient

Disadvantage: Expensive iterations (requires game re-solve on each iteration)

Zero-Sum LP-Based Algorithm

Repeat T times:

- **Solve** both the upper- and lower-bound pseudogames *exactly* (with e.g., an LP solver)
- **Sample** one play-through from the optimistic profile (in which the max-player her equilibrium strategy in the upper-bound game, and the min-player plays her strategy in the lower-bound game)
- **Create** the next pseudogame:
  - **Expand** all encountered nodes
  - **Update** empirical nature distributions of nature nodes sampled during play

Output: Pessimistic profile, and ε_T = difference in values between upper- and lower-bound pseudogames

Advantage: Sample-efficient

Disadvantage: Expensive iterations (requires game re-solve on each iteration)

Zero-Sum LP-Based Algorithm

Repeat T times:

- **Solve** both the upper- and lower-bound pseudogames *exactly* (with e.g., an LP solver)
- **Sample** one play-through from the optimistic profile (in which the max-player her equilibrium strategy in the upper-bound game, and the min-player plays her strategy in the lower-bound game)
- **Create** the next pseudogame:
  - **Expand** all encountered nodes
  - **Update** empirical nature distributions of nature nodes sampled during play

Output: Pessimistic profile, and ε_T = difference in values between upper- and lower-bound pseudogames

Theorem: The *best iterate* of the algorithm converges at rate

$$\mathbb{E}\varepsilon_T \leq \tilde{O}\left(\frac{N_T}{\sqrt{T}}\right)$$

number of nodes in final pseudogame (may be $\ll$ total number of nodes!)

Regret-Based Algorithm

Idea: Just use a regret minimizer, like CFR, for each player

Regret-Based Algorithm

Repeat T times:

- **Query** the regret minimizers for all players to obtain a strategy profile
- **Sample** one play-through from that strategy profile
- **Pass** the *optimistic* rewards to the regret minimizers
- **Create** the next pseudogame:
  - **Expand** all encountered nodes
  - **Update** empirical nature distributions of nature nodes sampled during play

Output: Average strategy profile

Several problems!

Regret-Based Algorithm

Repeat T times:

- **Query** the regret minimizers for all players to obtain a strategy profile
- **Sample** one play-through from that strategy profile
- **Pass** the *optimistic* rewards to the regret minimizers
- **Create** the next pseudogame:
  - **Expand** all encountered nodes
  - **Update** empirical nature distributions of nature nodes sampled during play

Output: Average strategy profile

Problem 1: The strategy space of each player is changing over time

Solution: CFR "handles it naturally". *Formalization*: "Extendable" regret minimizers

Regret-Based Algorithm

Repeat T times:

- **Query** the regret minimizers for all players to obtain a strategy profile
- **Sample** one play-through from that strategy profile
- **Pass** the *optimistic* rewards to the regret minimizers
- **Create** the next pseudogame:
  - **Expand** all encountered nodes
  - **Update** empirical nature distributions of nature nodes sampled during play

Output: Average strategy profile

Problem 2: We don't want to run a full CFR iterate on every sample; that is expensive

Solution: Use MCCFR + outcome sampling. Still works.

Regret-Based Algorithm

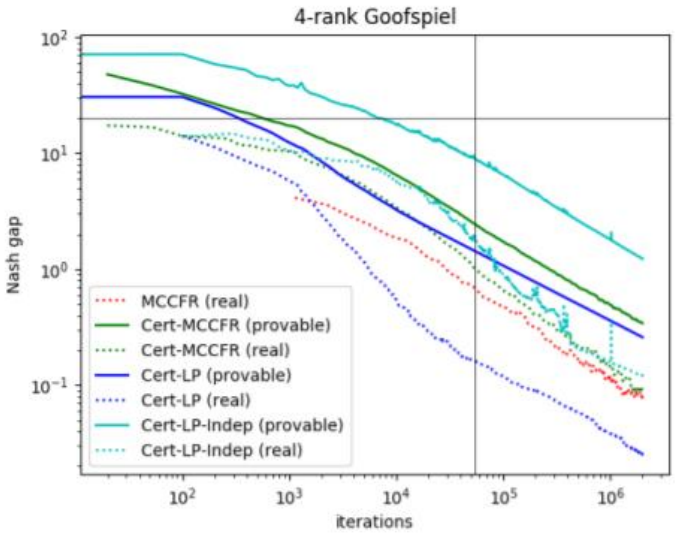
Repeat T times:

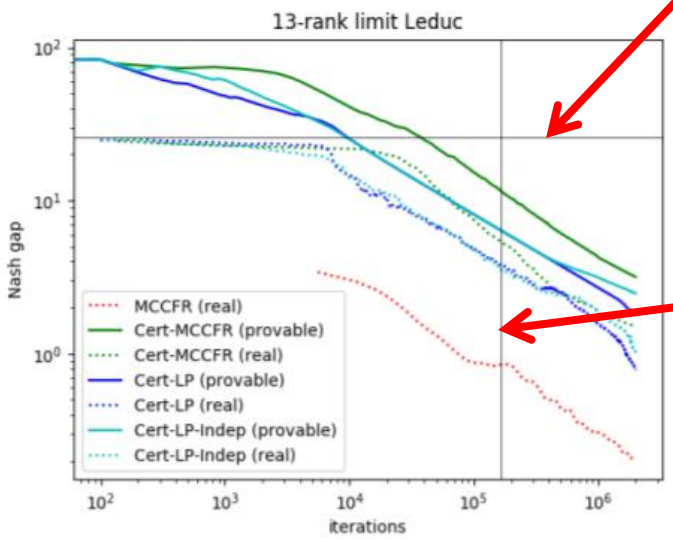
- **Query** the regret minimizers for all players to obtain a strategy profile
- **Sample** one play-through from that strategy profile
- **Pass** the *optimistic* rewards to the regret minimizers
- **Create** the next pseudogame:
  - **Expand** all encountered nodes
  - **Update** empirical nature distributions of nature nodes sampled during play

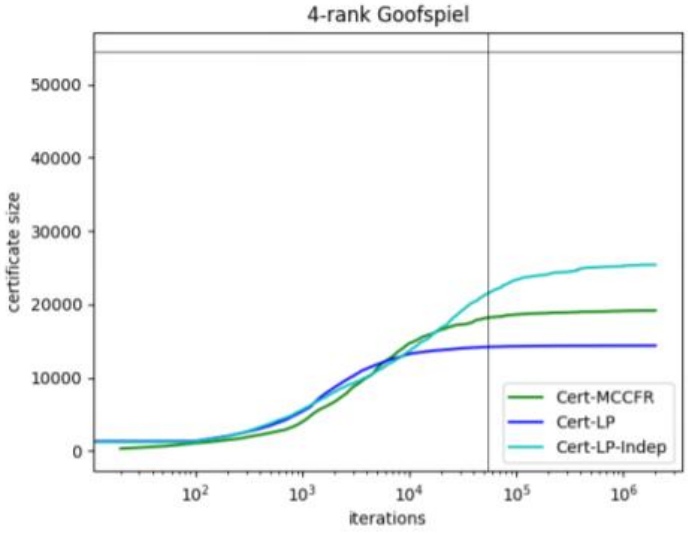
Output: Average strategy profile

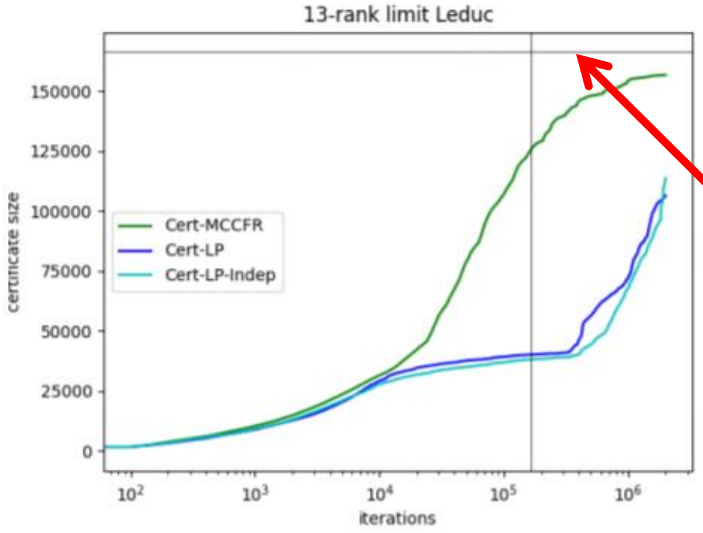
Problem 3: What equilibrium gap bound can we compute?

Experiments







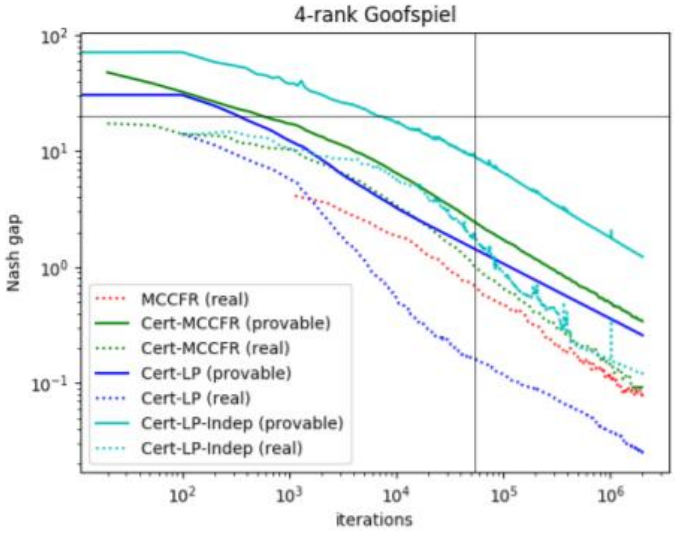


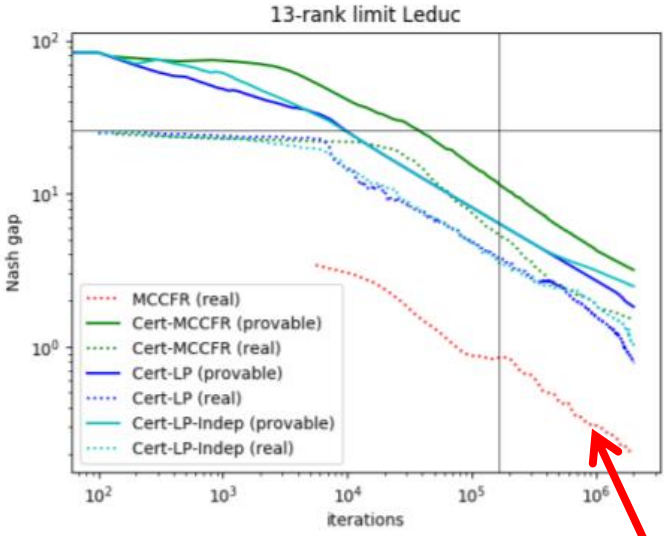
Horizontal line: reward bound of full game

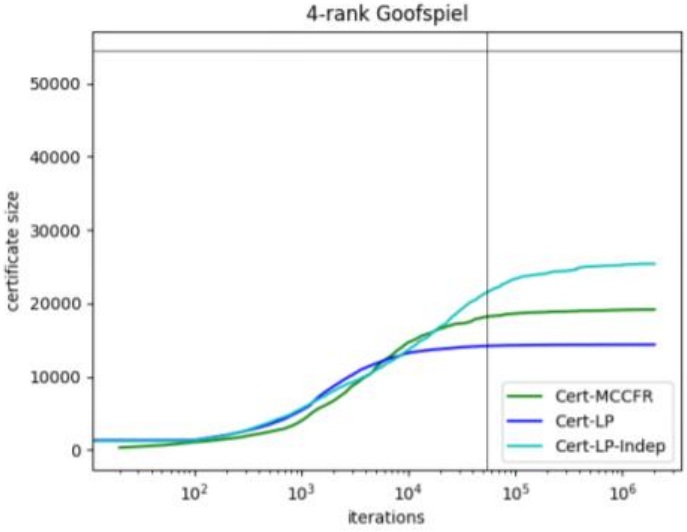
Vertical line: number of nodes in full game

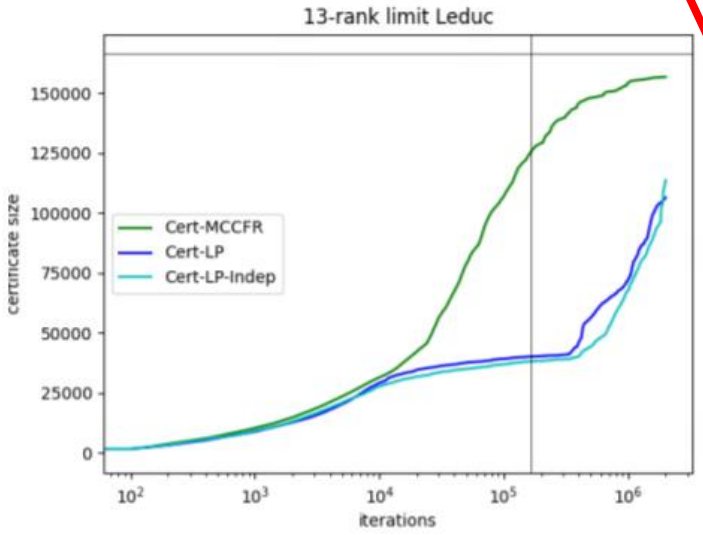
Horizontal line: number of nodes in full game

Experiments





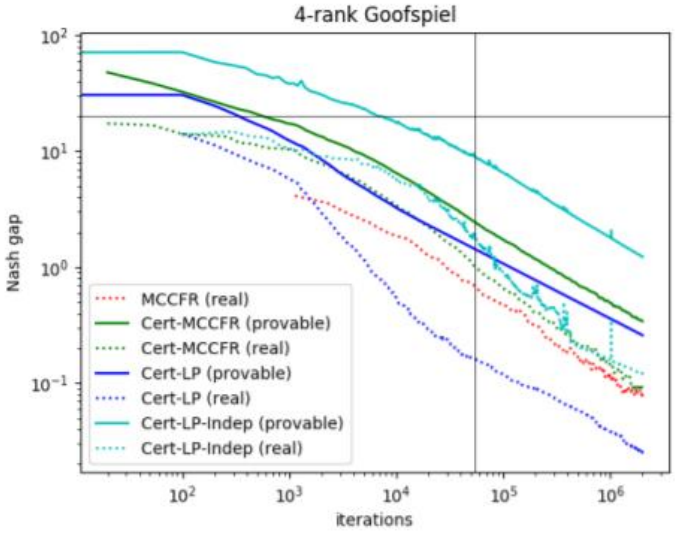


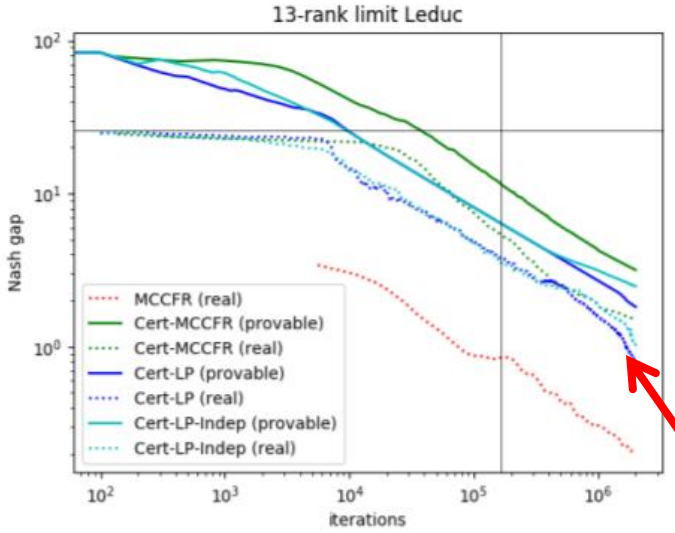


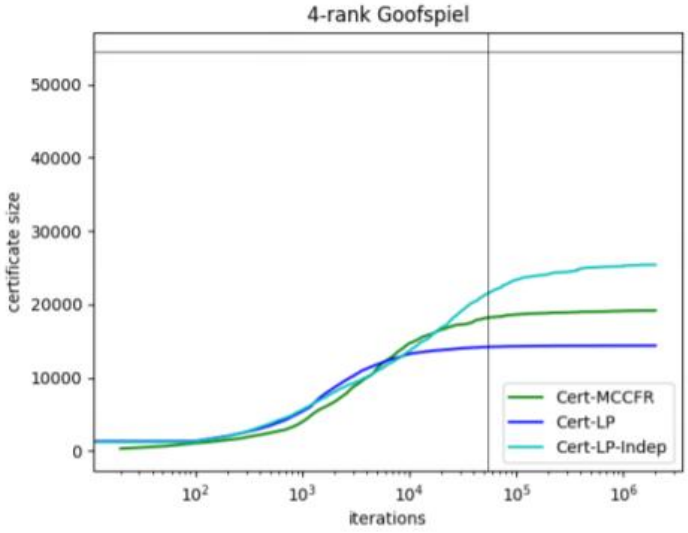
In all games, with all algorithms, nontrivial certificates are found **without expanding the full game tree,** in fact, **with fewer game samples than there are game tree nodes**

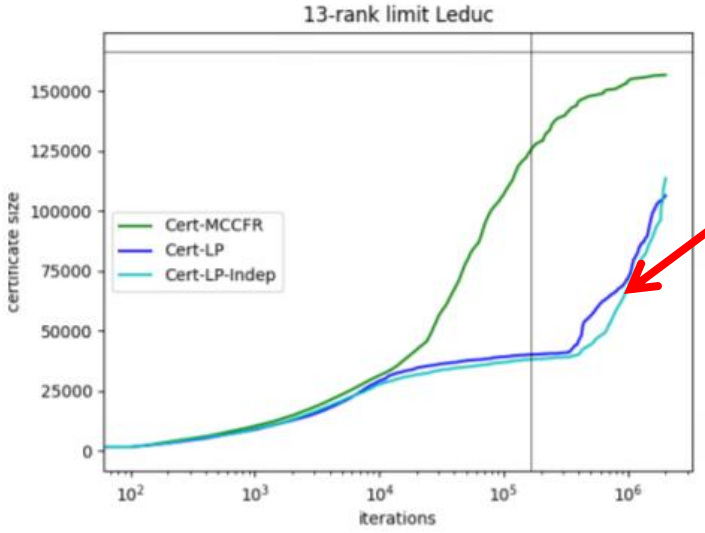
MCCFR converges quickly in reality, but this cannot be verified without expanding the rest of the game tree

Experiments









In all games, with all algorithms, nontrivial certificates are found **without expanding the full game tree,** in fact, **with fewer game samples than there are game tree nodes**

LP-based certificate finding has better sample efficiency and final certificate size than regret-based, but (not shown) runs slower