What if we don't know the game? Finding and Certifying (Near-)Optimal Strategies in Black-Box Extensive-Form Imperfect-Information Games

Tuomas Sandholm

Carnegie Mellon University

Strategic Machine, Inc.

Strategy Robot, Inc.

Optimized Markets, Inc.

Joint work with my PhD student **Brian Zhang**
[NeurIPS-20, AAAI-21 & draft]

OPTIMIZED MARKETS

STRATEGY ROBOT, INC.

STRATEGIC MACHINE, INC.

Carnegie Mellon University Computer Science Department

There has been amazing progress in game solving over the last 16 years.

Strategy Robot, Inc.

Strategic Machine, Inc.

What if the game model is inaccurate or unknown?

1. Lossy game abstraction techniques with ε-exploitability guarantees [S. & Singh, EC-12; Kroer & S., EC-14, AAMAS-15, EC-16, NeurIPS-18] apply to modeling also

2. **THIS TALK:** First techniques for computing provably (near-)equilibrium strategies while searching only a tiny fraction of the game tree [Zhang & S., NeurIPS-20, AAAI-21]
   - => algorithm with optimal $\tilde{O}(\#\text{nodes}/\sqrt{T})$ convergence in this setting
   - Prior methods (such as MCCFR) can be exponential in tree size

Black-box games

- Game is not explicitly given in the form of rules, but rather via access to playing it
  - We can control all players during the practice phase
- E.g., war games, strategy video games, and financial simulations

STRATEGY ROBOT, INC.

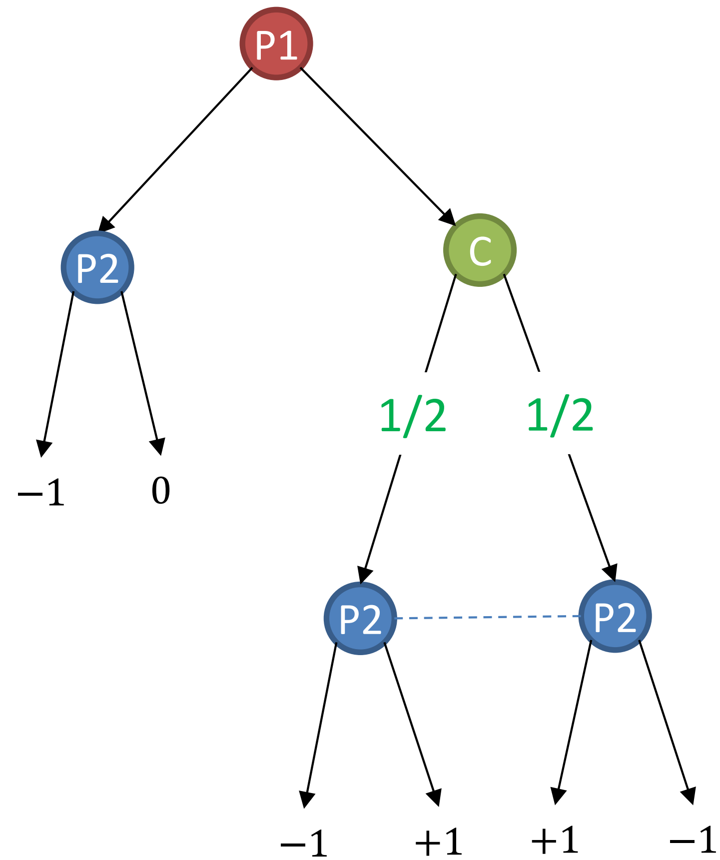
Learning to play black-box games

- **Deep Reinforcement Learning** (e.g., *AlphaStar* [Vinyals et al., 2019], *OpenAI Five* [Berner et al., 2019])
  - Strong practical performance for a while
  - **Issue:** No exploitability bounds
    - Leads to strategies that can be beaten in practice also


- **Bandit Regret Minimization** [Farina & Sandholm, AAAI-21]
  - Converges to ε-equilibrium after poly(N, $1/\varepsilon$) game samples (N = size of game)
  - **Issues (online MCCFR [Lanctot et al. 2009] has these issues also and other issues):**
    - Worst-case exploitability bounds are trivial until number of iterations is much larger than N
    - To compute *ex post* exploitability guarantee, would need to expand rest of game tree

- **This talk** [Zhang & Sandholm, NeurIPS-20, AAAI-21]
  - Compute Nash equilibrium by incrementally expanding the game tree
  - Exploitability bounds always computable *ex post* without expanding remainder of tree!

Extensive-form games

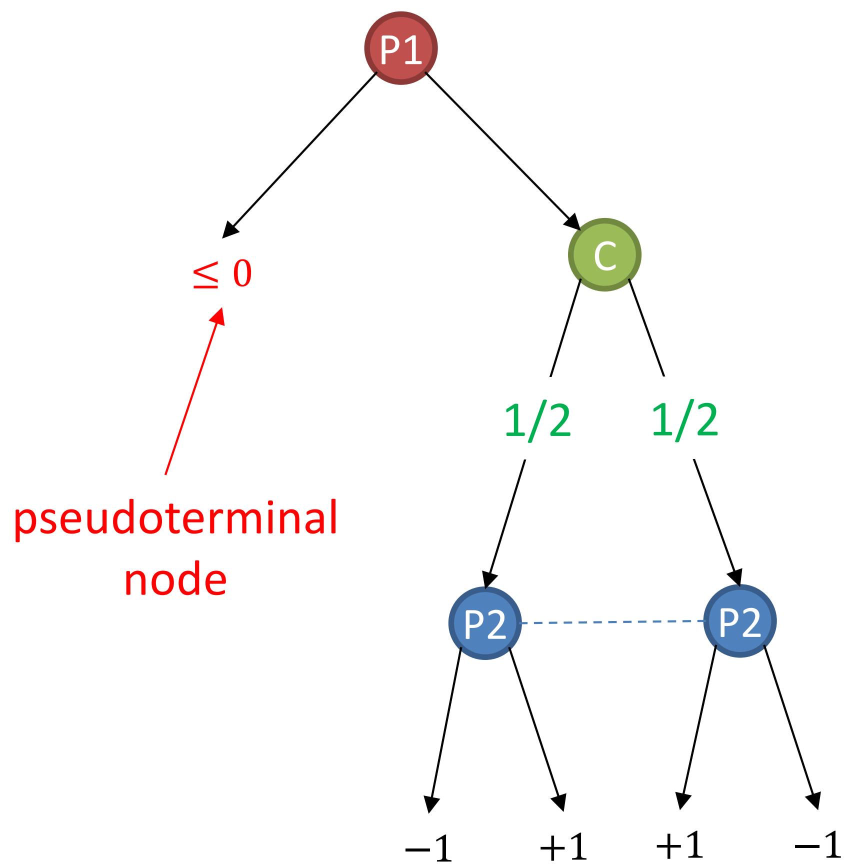


Pseudogames and certificates

Pseudogame: Partially-expanded game without known utilities on all terminal nodes

In 0-sum setting, gives rise to **two games**:

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

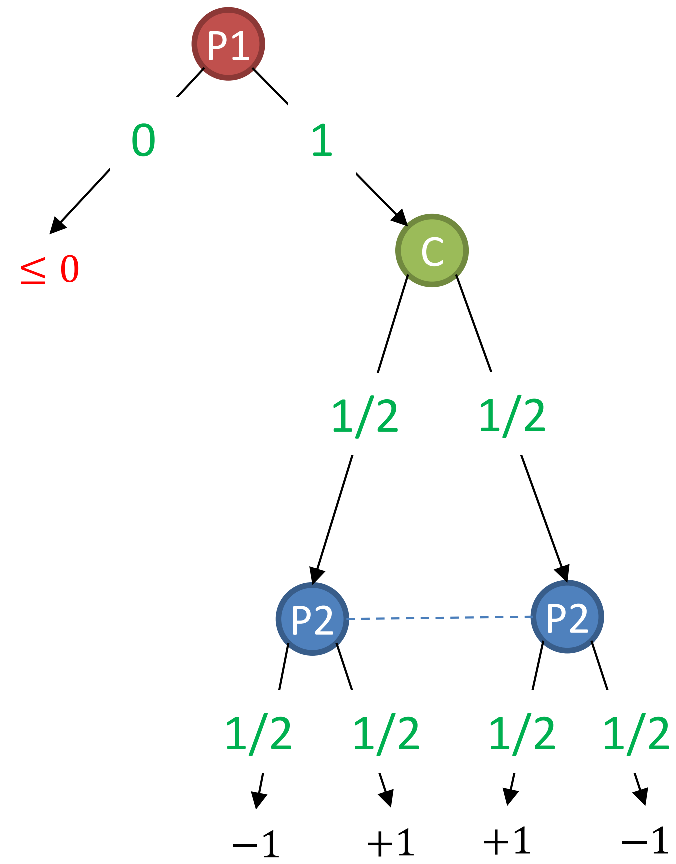


Pseudogames and certificates

ε-Nash equilibrium in a pseudogame: strategy profile in which every player is *provably* playing an ε-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 game and (ε-)Nash equilibrium of that pseudogame



Small certificates

Question: When do small ε-Nash certificates exist?
Specifically, size $O\left(N^c \mathrm{poly}(1/\varepsilon)\right)$ for some $c < 1$

Again, N is the number of nodes

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 (e.g., 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

	A	B	C	D	E	F	G
T							
U							
V							
W							
X							
Y							
Z							

Small certificates

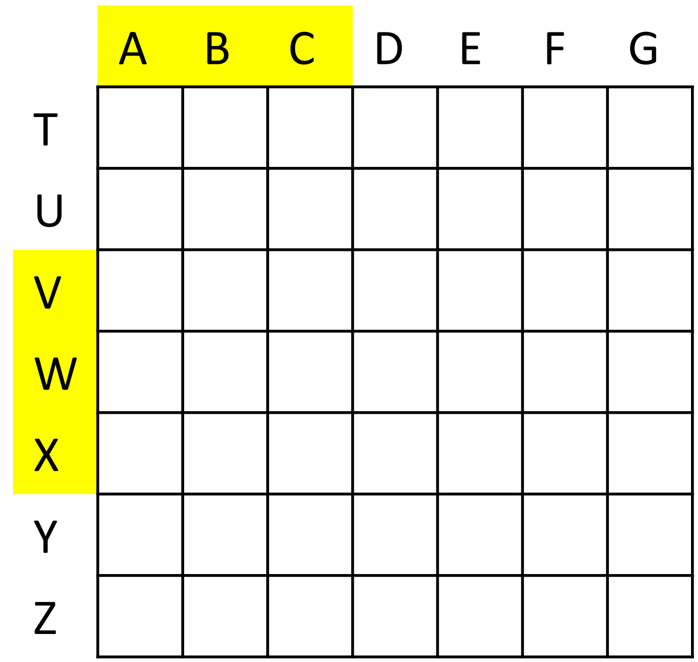
Answer #2: They exist in (squarish) **normal-form games**

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

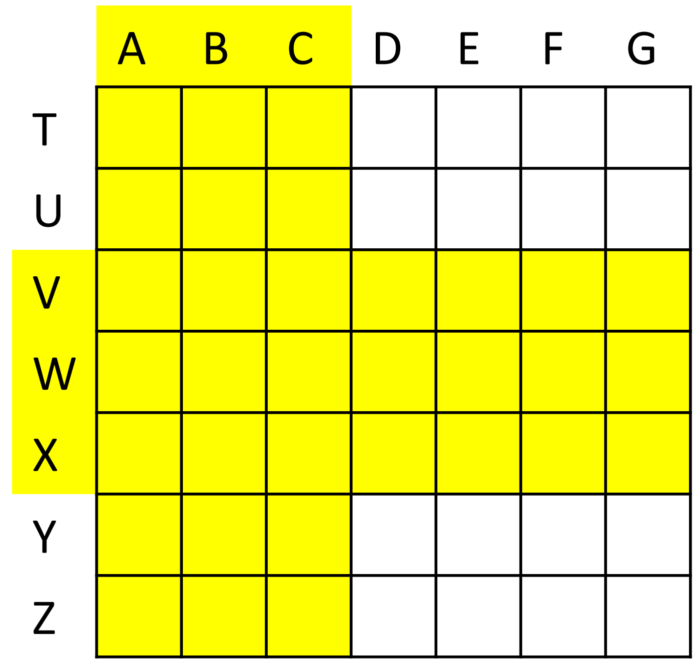


Small certificates

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

Proof:

We only need those rows and columns!



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

Small certificates don't always exist

Counterexample: Consider this game:

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

Game tree size: 4^T

Theorem: Any ε-certificate of this game must have size

$$\Omega\left(4^{T(1-2\varepsilon)}\right)$$

Proof sketch: P1's strategy **must have high entropy**, but this is not possible unless lots of nodes get expanded

More bad news

Theorem: It is NP-hard to approximate the smallest certificate of an extensive-form 0-sum game to better than an $O(\log N)$ multiplicative factor

Proof idea: Reduction from set cover

A generous oracle model to start…

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

- Upper and lower bounds (maybe not tight) 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 chance, the exact chance 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

Yet more bad news

Theorem: With only an oracle for an extensive-form 0-sum game, there is no equilibrium-finding algorithm that runs in time polynomial in the size of the smallest certificate

Proof: One-player “guess log(N) bits one by one” game:
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* (e.g., using an LP solver)
- **Create** the next pseudogame by expanding all pseudoterminal nodes in the support of the **optimistic profile** (i.e., profile in which the max-player plays 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* (e.g., using an LP solver)
- **Create** the next pseudogame by expanding all pseudoterminal nodes in the support of the **optimistic profile** (i.e., profile in which the max-player plays 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* (e.g., using an LP solver)
- **Create** the next pseudogame by expanding all pseudoterminal nodes in the support of the **optimistic profile** (i.e., profile in which the max-player plays 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 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	—

Realistic oracle (e.g., simulator)

Assume access to a **simulator:**

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

Goal:

- *Compute and verify "ex-post"* approximate equilibria
- 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 bound

Theorem: Consider any algorithm with the following guarantee.

For some constant $\gamma > 0$,

given a 0-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, our pseudogame uses the empirical distribution given by the samples, but in addition, to represent uncertainty, we give each player a reward $[-\rho, \rho]$, where

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

range of utilities possible from h

times h has been reached

confidence parameter

Choice of confidence bound

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

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

NEW IDEA SINCE OUR AAAI-21 PAPER:

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

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

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

Main tool:
Pseudogames as confidence bounds

Confidence bounds are actually 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

LP-based algorithm for 0-sum games

Repeat T times:

- **Solve** both the upper- and lower-bound pseudogames *exactly* (e.g., using an LP solver)
- **Sample** one play-through from the optimistic profile
- **Create** the next pseudogame:
  - **Expand** the first encountered node not already in the pseudogame
  - **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

Connections to what was known:

- In perfect-information game with no nature randomness, it's **alpha-beta search**
- In the one-player "multi-armed bandit" setting, it's **UCB**
  - (except algorithm has a different constant in the upper confidence bound term, and so does the regret bound)

LP-based algorithm for 0-sum games

Advantage: Sample-efficient

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

- We warm start from the previous LP, whose values typically change very little based on the one new sample

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

Regret-based algorithm
(can also be used for coarse correlated equilibrium in general-sum games)

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** each player's regret minimizer that player's *optimistic* reward
- **Create** the next pseudogame:
  - **Expand** the first encountered node not already in the pseudogame
  - **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** each player's regret minimizer that player's *optimistic* reward
- **Create** the next pseudogame:
  - **Expand** the first encountered node not already in the pseudogame
  - **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** each player's regret minimizer that player's *optimistic* reward
- **Create** the next pseudogame:
  - **Expand** the first encountered node not already in the pseudogame
  - **Update** empirical nature distributions of nature nodes sampled during play

Output: Average strategy profile

Problem 2: Running a full CFR iterate on every sample would be expensive

Solution: Use MCCFR with outcome sampling. Nothing breaks

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** each player's regret minimizer that player's *optimistic* reward
- **Create** the next pseudogame:
  - **Expand** the first encountered node not already in the pseudogame
  - **Update** empirical nature distributions of nature nodes sampled during play

Output: Average strategy profile

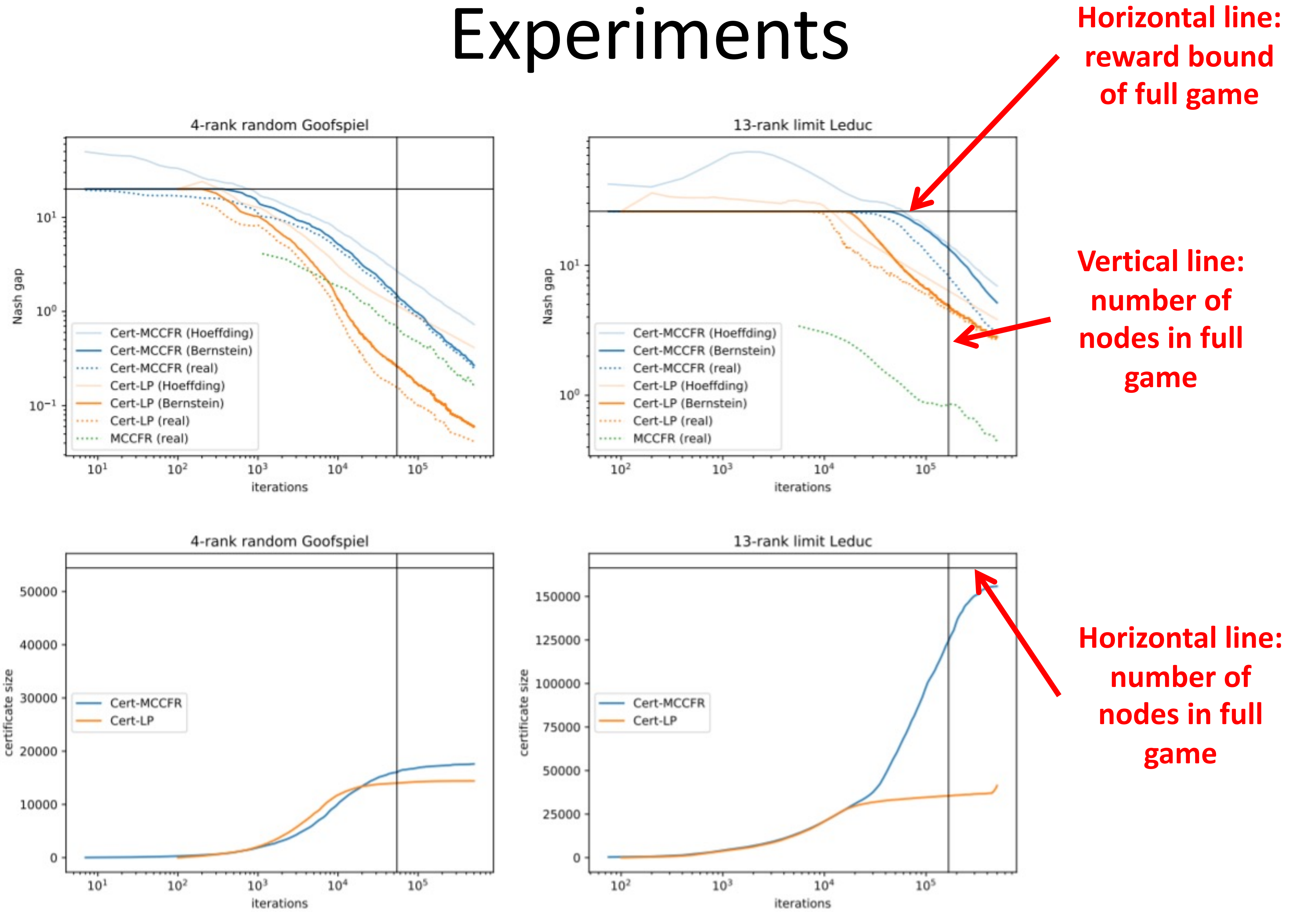
Problem 3: What equilibrium gap bound can we compute?

What equilibrium gap bound can we compute?

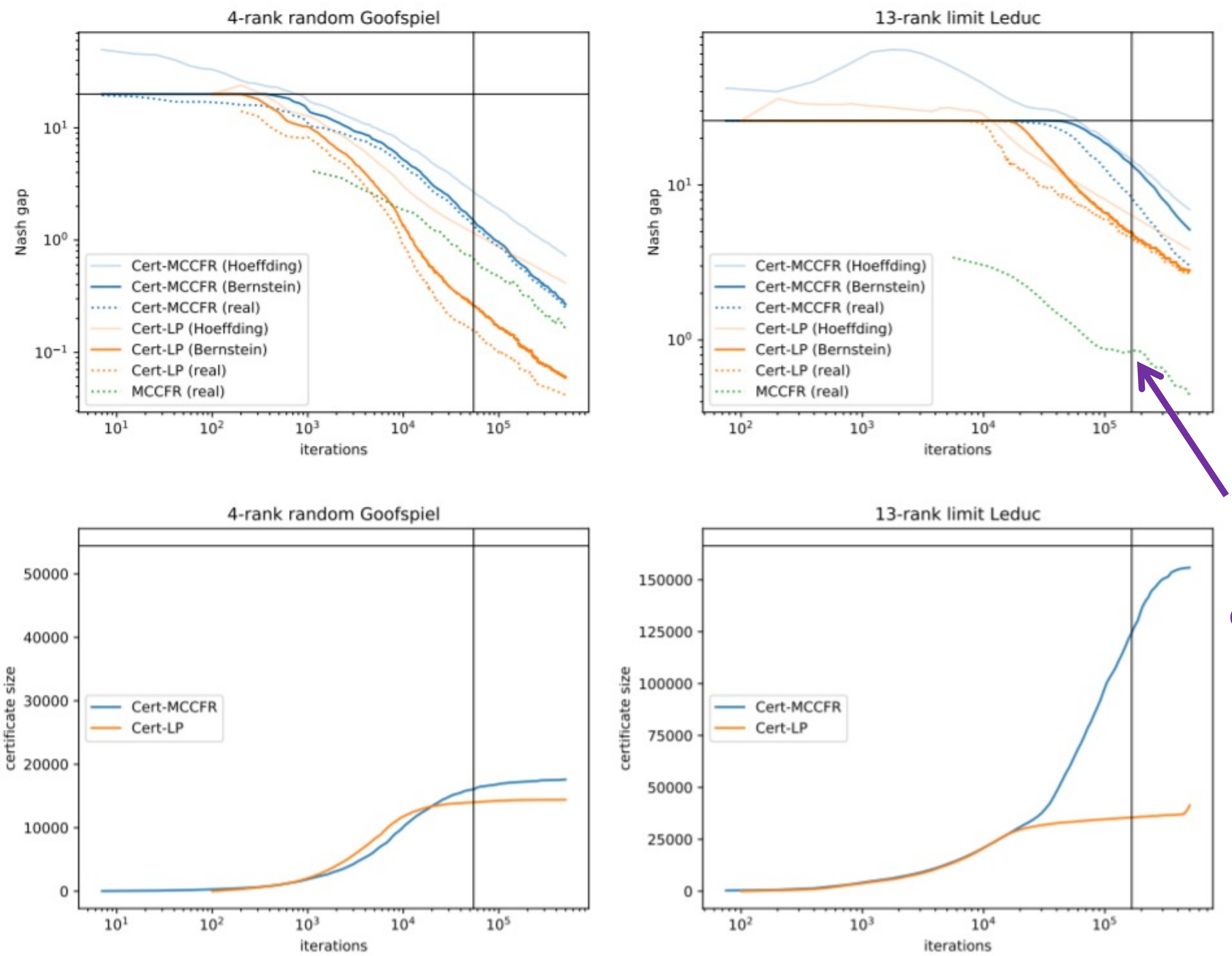
- The natural game-specific equilibrium gap bound —used in our exact LP-based algorithm—(difference in optimistic best response values using the final pseudogame) **doesn't converge as $\tilde{O}(1/\sqrt{T})$** in the worst case
- …but, we know that the *worst-case-over-games* equilibrium gap bound of the algorithm *does* converge as $\tilde{O}(1/\sqrt{T})$ (for the same reason that MCCFR does)

- **Solution**: In practice, take the former; it's basically always smaller. In theory, take the minimum of the two

Experiments



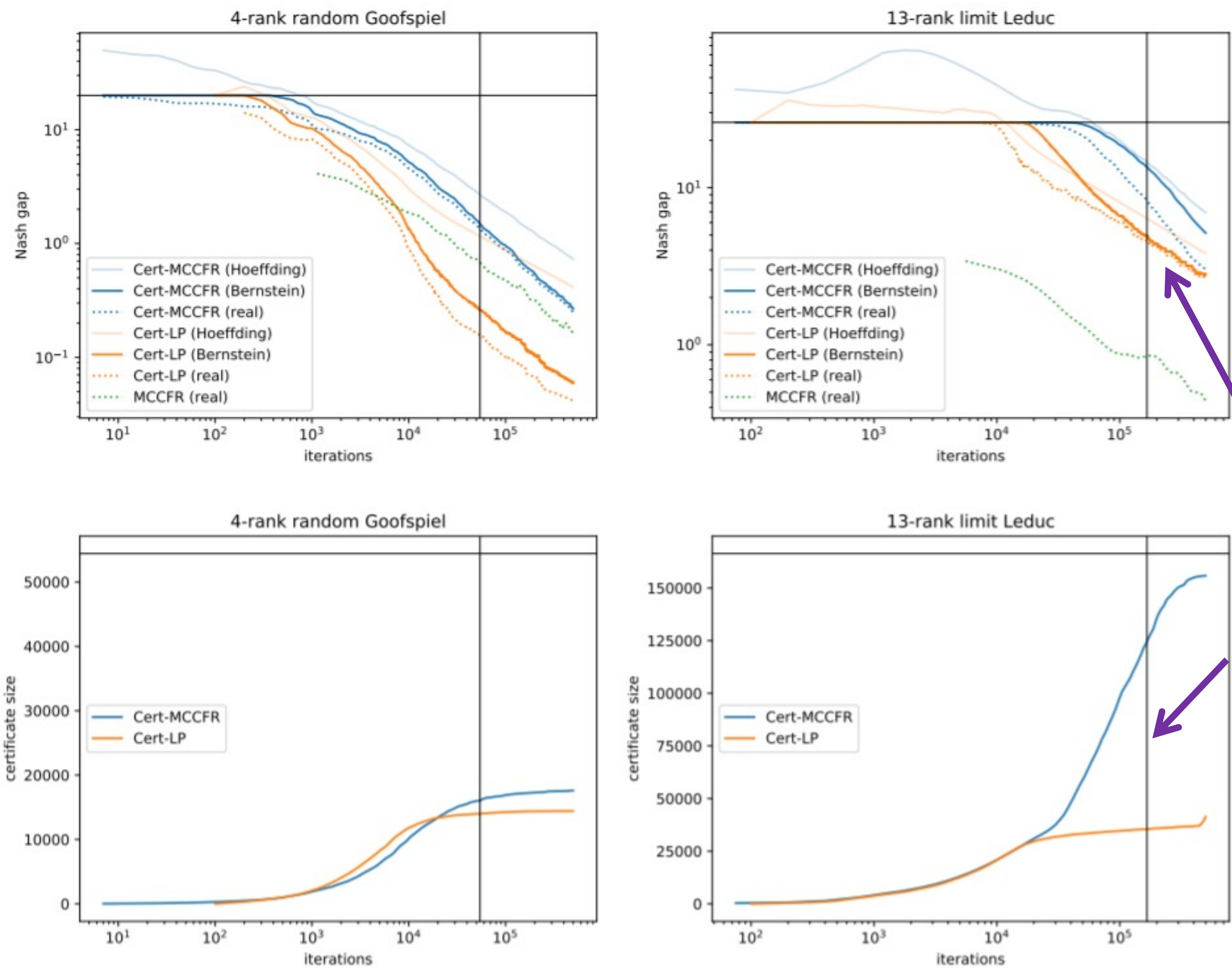
Experiments



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

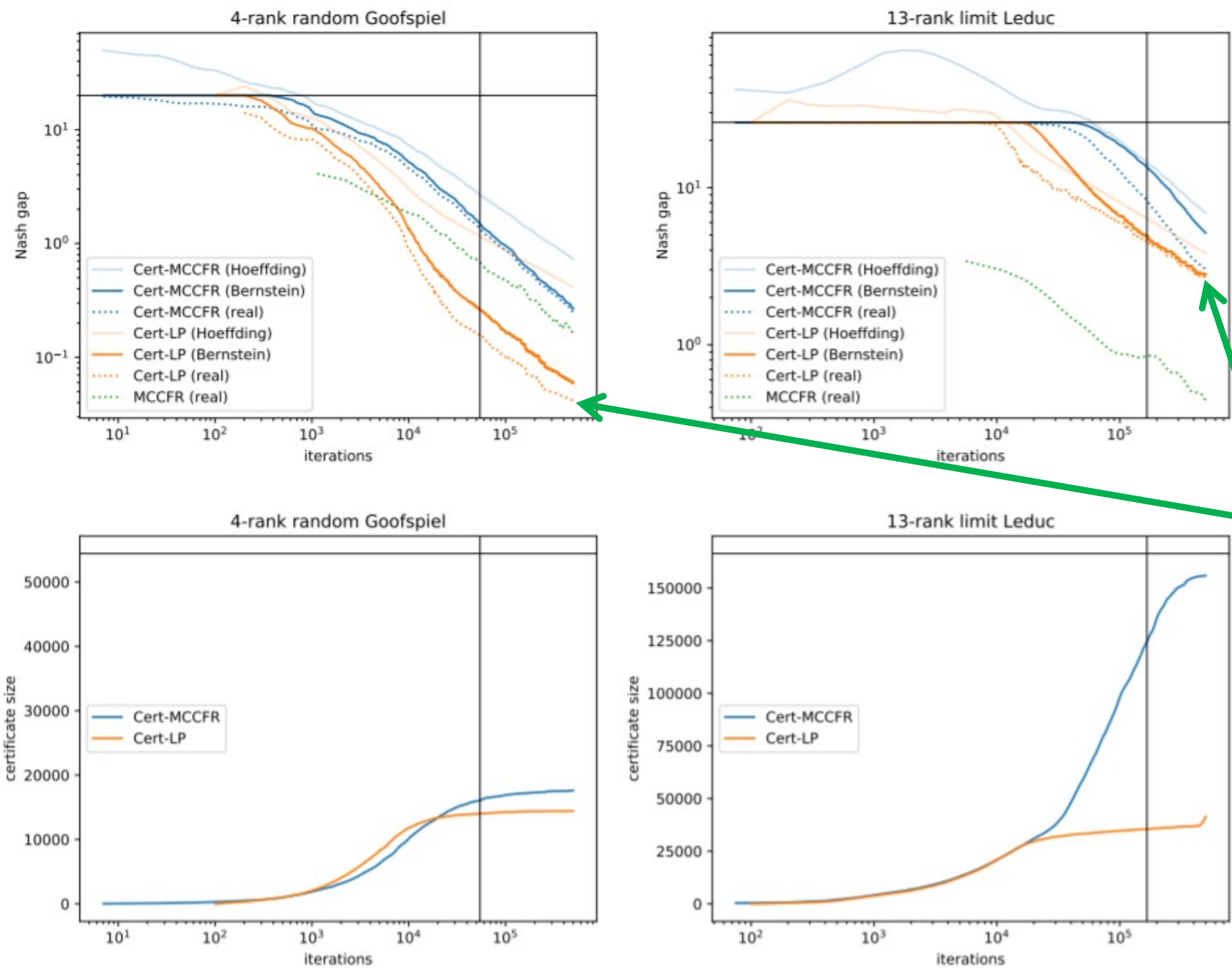
Experiments



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

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**

Bernstein gives tighter equilibrium gap bounds—nearly perfectly tight in most cases

Conclusion

- Black-box imperfect-information games (of at least moderate size) can now be **solved**
  - i.e., get the non-exploitability guarantee of game theory
- This talk covered parts of the following papers and a new concentration result
  - Finding and Certifying (Near-)Optimal Strategies in Black-Box Extensive-Form Games, *AAAI*-21 https://arxiv.org/abs/2009.07384
  - Small Nash Equilibrium Certificates in Very Large Games, *NeurIPS*-20 https://arxiv.org/abs/2006.16387