Algorithmic Game Theory

DeepStack

Viliam Lisý

Artificial Intelligence Center
Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University in Prague

(May 25, 2018)

Home exercise

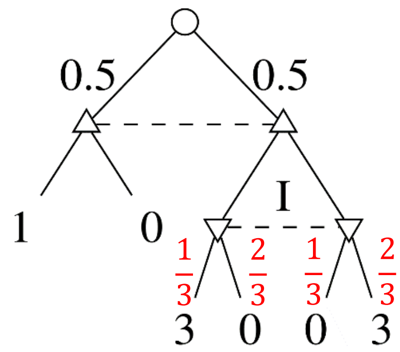
Prove or disprove the following conjecture.

Conjecture: For any n-player game and each $\epsilon > 0$, no-regret dynamic eventually converges to ϵ –Nash equilibrium.

Game decomposition

Perfect information example

Imperfect information example



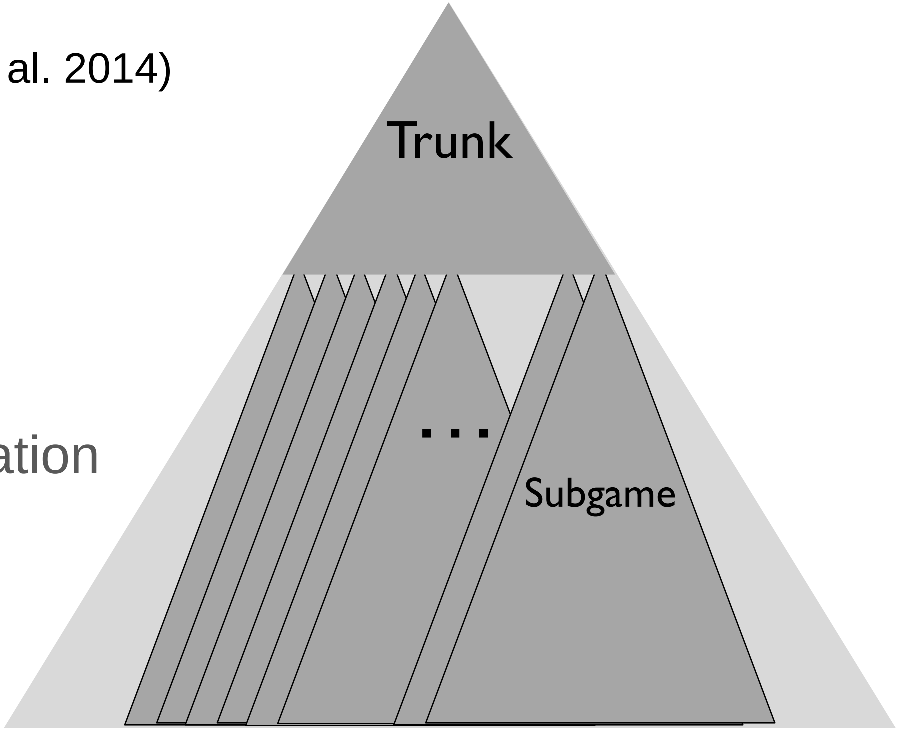
CFR-D

CFR with Decomposition (Burch et al. 2014)

Trades-of space for computation

- Store only the trunk
- Resolve subgames in each iteration

Resolve on demand in play



CDR-D

Augmented information set

- Set on undistinguishable histories for any player, not just the deciding one

Subgame (denoted S)

- forest of trees closed under descendance and belonging into augmented information sets

R(S)

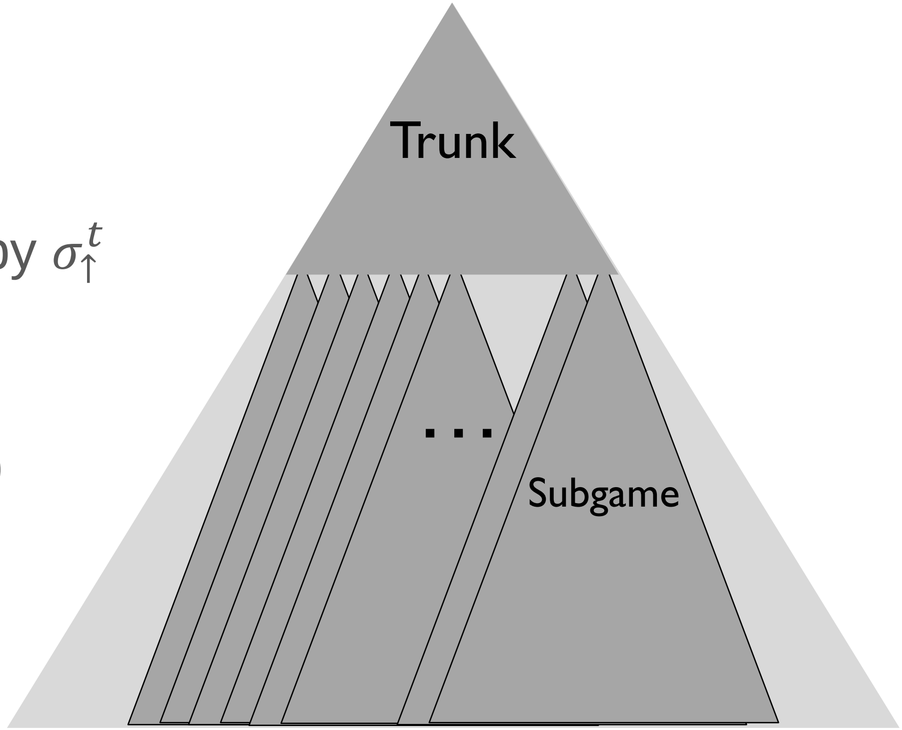
- set of augmented information sets in the root of a subgame

CFR-D: Solving Trunk Strategy

Initialize regrets to 0

For iteration $t = 1, \ldots, T$

- compute $\sigma_\uparrow^t$ from stored regrets
- update trunk average strategy by $\sigma_\uparrow^t$
- For each subgame S
  - $\sigma_S^t \leftarrow$SOLVE(S, $\sigma_\uparrow^t$)
  - For each augmented $I_p \in R(S)$
    - Compute value v_{I_p}
    - Update average value $\mathrm{cf}v_{I_p}$
- Update trunk regrets using v_{I_p}



CFR-D: Computing Trunk Strategy

CFR-D: Resolving Subgame

Assume blue player played D and the game reached S1

Unsafe resolving

Save resolving

No incentive to change trunk!

CFR-D More Complicated Resolving

CFR-D Resolving Game

When resolving for player 1

- Create new chance node as the root
- Create new nodes for player 2 grouped by her “information sets”
- Connect the root to nodes in proportion to player 1 trunk strategy
- For each player 2 node, add follow action leading to subgame
- For each player 2 node, add terminate action with CFV of IS

We need

- Distribution in the root IS generated by player 1 trunk strategy
- Counterfactual value achievable by player 2 in his root ISs

CFR-D Convergence properties

CFR-D achieves no regret in the trunk

It the counterfactual regret at each information set I at the root of a subgame is bounded by ϵ_S, then than the average regret over the whole game is

$$R^T_{full} \leq \frac{N_{TR}\sqrt{A}}{\sqrt{T}} + N_S \epsilon_S$$

Proof sketch: $\sigma^0[S \leftarrow \sigma_S^{0,*}], \sigma^1[S \leftarrow \sigma_S^{1,*}], \ldots$

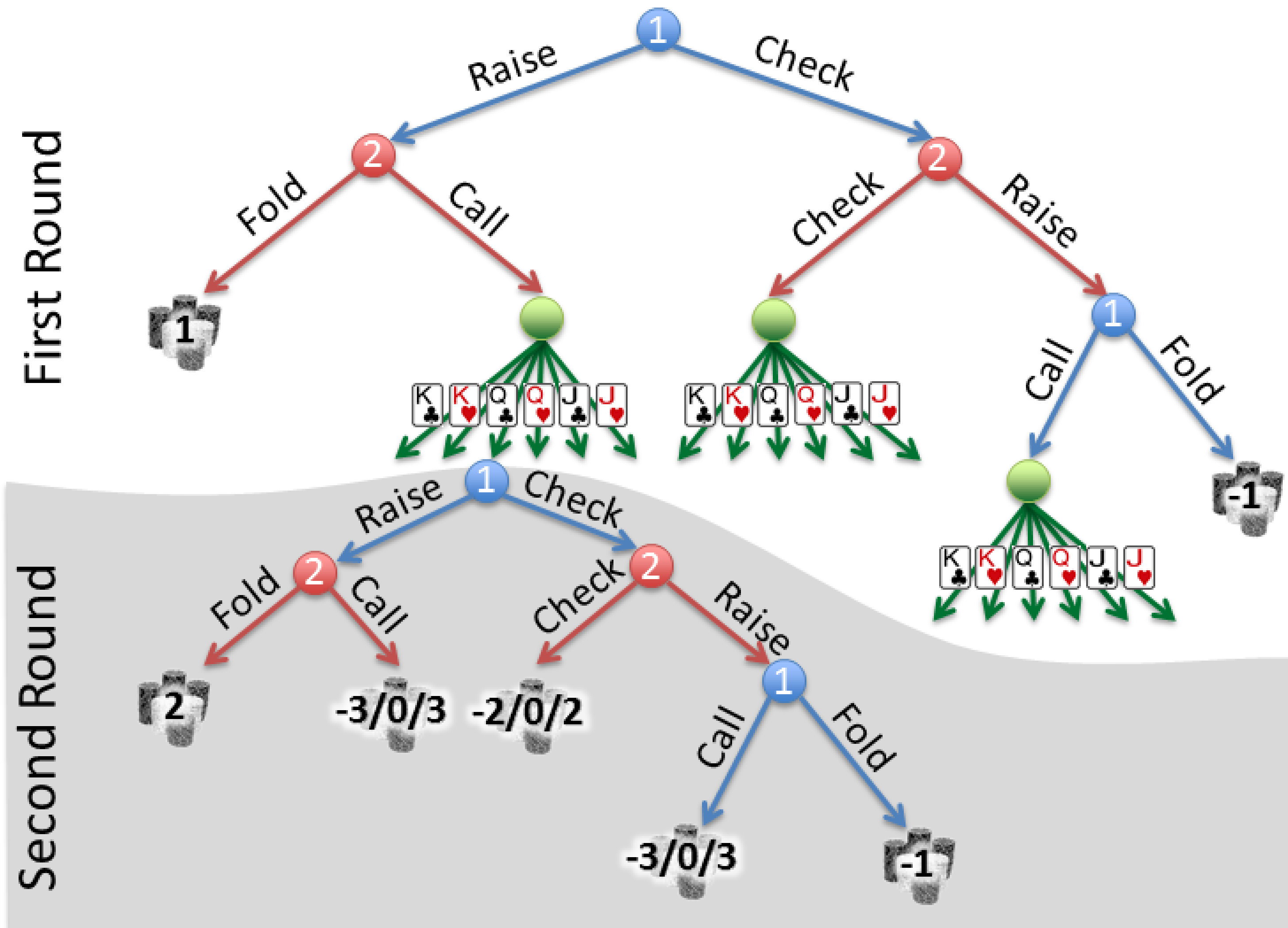
- CF regret in the trunk minimized by CFR
- CF regret in the subgame close to 0 for both players

CFR-D resolving forms a Nash equilibrium

If we run the recovery game for each player and each subgame until we reach regret below ϵ_R, the combined strategy has regret

$$R^T_{full} \leq \frac{N_{TR}\sqrt{A}}{\sqrt{T}} + N_S(3\epsilon_S + 2\epsilon_R)$$

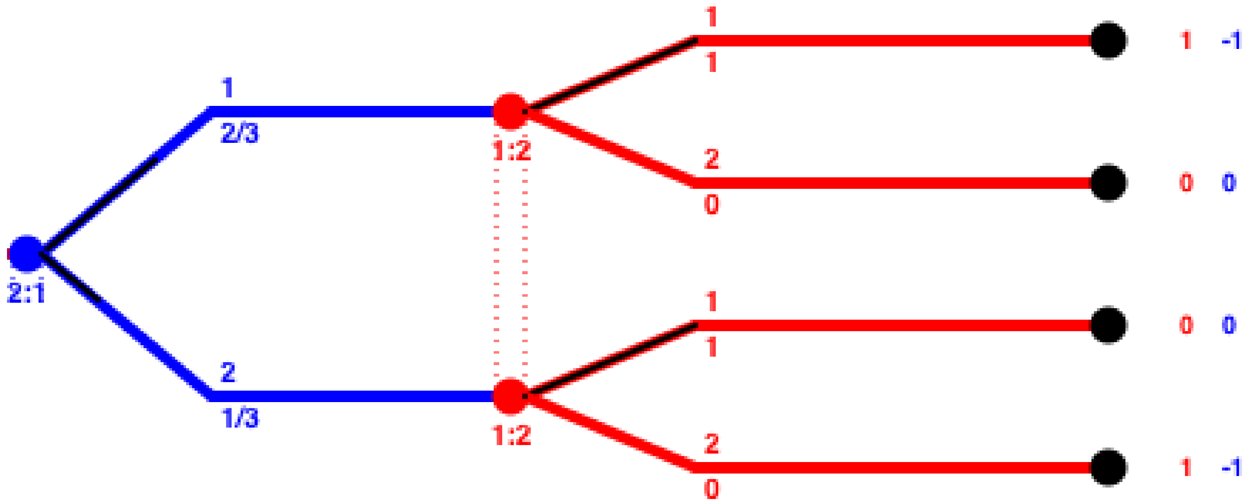
Public Tree



First Round
Second Round
1
Raise
Check
2
Fold
Call
Check
Raise
1
K Q J
Call
Fold
-1
Raise
Check
Fold
Call
Check
Raise
2
-3/0/3
-2/0/2
Call
Fold
-3/0/3
-1

Public Tree

Matching pennies



Phantom Tic-Tac-Toe

Visibility-based pursuit-evasion games

Augmented IS in poker public node

Resolving poker subgame

To resolve, we need

$\forall I_1 \in R(S) \;\; \pi_1(I_1)$

$\forall I_2 \in R(S) \;\; cfv_2(I_2)$

In poker it means

$\pi_1(I_1)$ - probability that player 1 holds each hand = range

$cfv_2(I_2)$ - how much player 2 can win with each hand

In root (after chance reals hole cards)

$\pi_i(I_i)$ - uniform

$cfv_i(I_i)$ - pre-computed offline

DeepStack: updating maintained values

Assuming DeepStack is player 1

Own action

- replace player 2's *cfv*s by the once computed in the resolve game
- update player 1's range based on the played strategy

Chance action

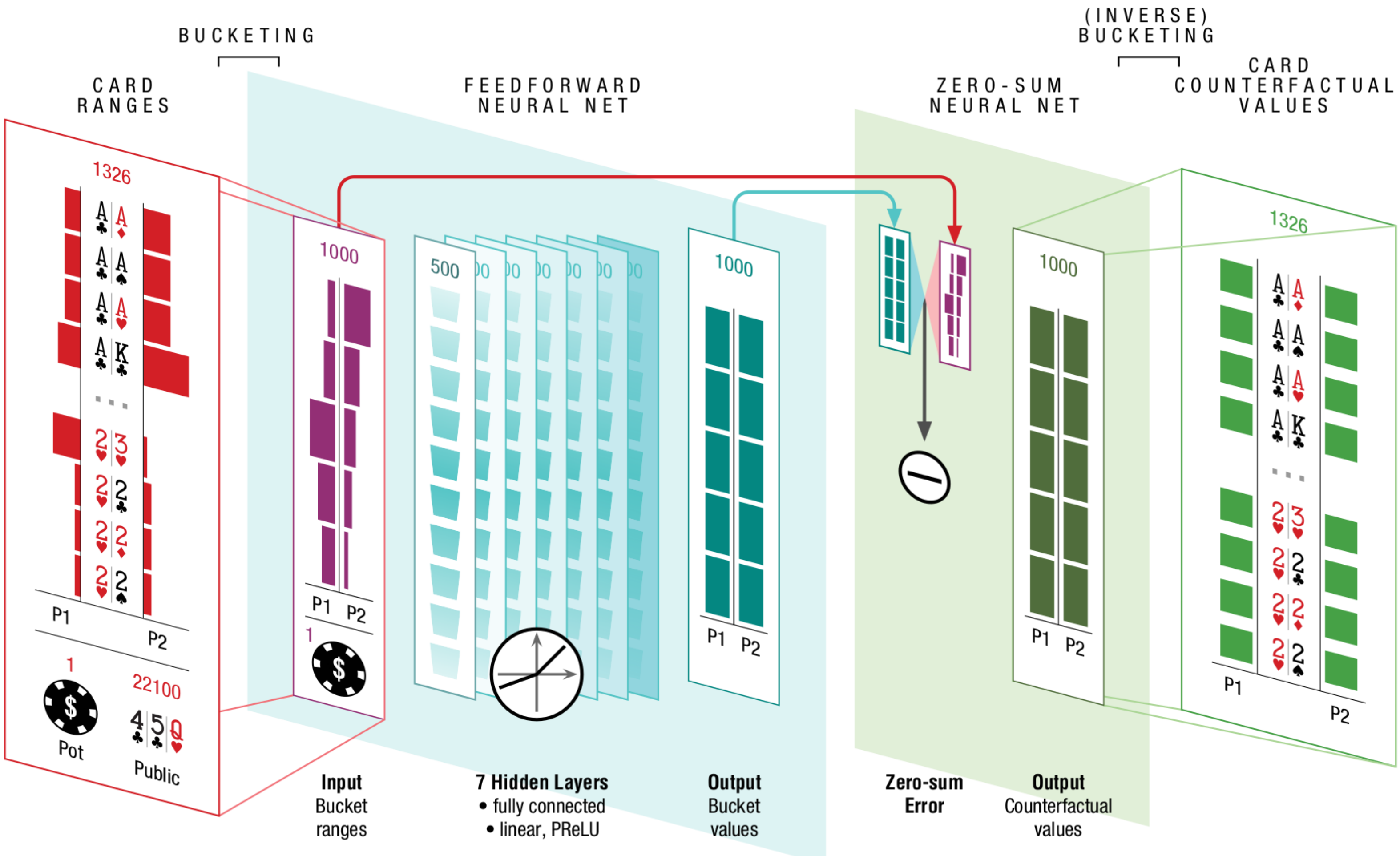
- replace player 2's *cfv*s from the last resolve above chance
- keep player 1's range unchanged

Opponent's action

- no update required!

DeepStack: Limited look-ahead

DeepStack: Neural Network



DeepStack: Training

Turn Network (right after dealing turn card)

- 10M pseudo-random ranges, pots, random boards
- Solve by CFR^{+} until the end of the game
- Extract CFVs for training, train Turn NN

Flop Network (right after dealing flop cards)

- 10M pseudo-random ranges, pots, random boards
- Solve by DeepStack (CFR-D) using the pre-trained Turn NN
- Extract CFVs for training, train Turn NN

Pre-flop Network

- 10M pseudo-random ranges, pots
- Enumerating 22100 possible flops and averaging

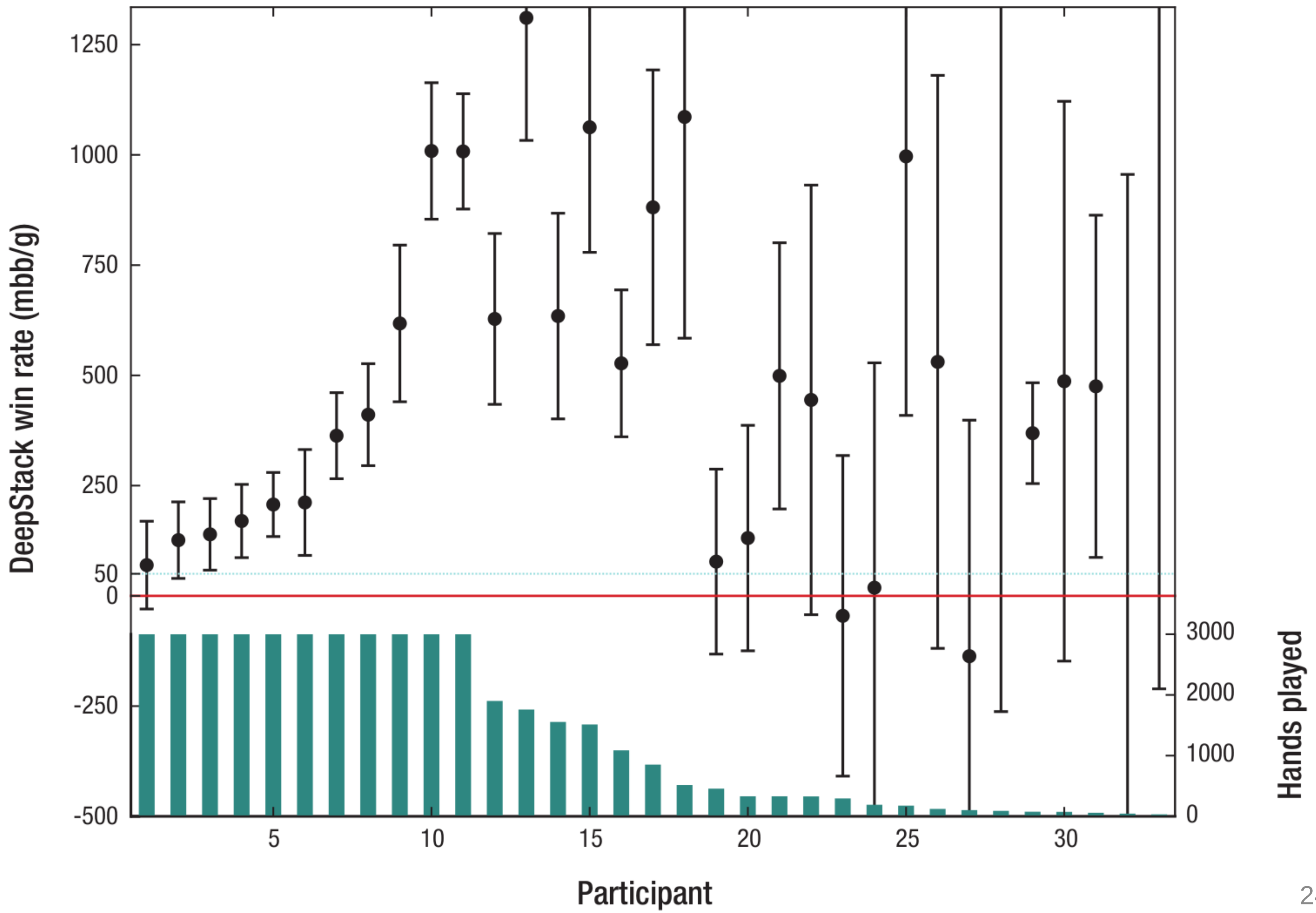
DeepStack: Convergence

Theorem: If the error of CFVs returned by the value function is less then ϵ and T iterations of resolving are used for each decision, than the exploitability of the player strategy is less than

$$k_1\epsilon + \frac{k_2}{\sqrt{T}}$$

where k_1, k_2 are game-specific constants.

DeepStack: Results



References

Burch, N., & Bowling, M. (2013). CFR-D: Solving Imperfect Information Games Using Decomposition. arXiv Preprint arXiv:1303.4441, 1–15. Retrieved from http://arxiv.org/abs/1303.4441

Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., Davis T., Waugh K., Johanson M., Bowling, M. (2017). DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker. https://doi.org/10.1126/science.aam6960