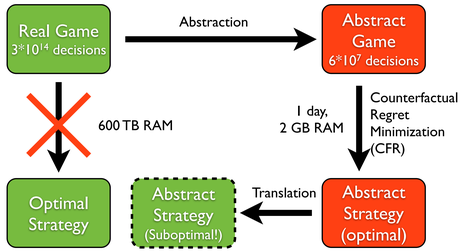
Finding Optimal Abstract Strategies in Extensive-Form Games

Michael Johanson, Nolan Bard, Neil Burch, Michael Bowling :: University of Alberta, Canada

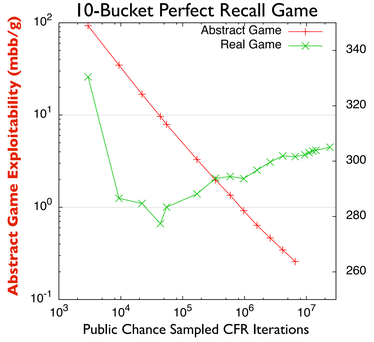
University of Alberta
Computer Poker Research Group

In two-player zero-sum games like poker, a **Nash equilibrium** strategy is **optimal** or **unexploitable**: on average, it will not lose to a worst-case opponent.

Although efficient algorithms exist for computing Nash equilibria, such as **Counterfactual Regret Minimization**, many games remain too large to solve. State-space **abstraction** techniques let us make the problem tractable.



CFR in an abstract 10-Bucket Perfect Recall Game



However - when both players use abstraction, the resulting strategy can be far from optimal in the real game:

Overfitting:
As CFR converges in the **abstract game**, it can get more exploitable in the **real game**.

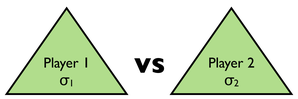
Abstraction Pathologies:
Intuitively, a larger, finer-grained abstraction should result in a less exploitable strategy. Unfortunately, even a strict refinement can result in higher exploitability!

Abstract Strategy Suboptimality:
An abstract game equilibrium might not be the abstract strategy with the lowest real game exploitability.

All of these problems are solved if the opponent is unabstracted, but that typically requires far too much memory.

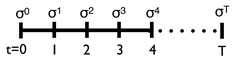
Counterfactual Regret Minimization (CFR), NIPS 2007

1. Initialize each player's strategy arbitrarily.



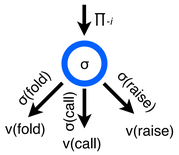
$\sigma = (\sigma_1, \sigma_2)$

2. Over T iterations, "play games" by walking the game tree. At any time, each player is using a **current strategy**.




```
for(t=0 to T) {
```

2a. Estimate payoffs of actions, Accumulate regret for actions, Create next iteration's strategy.



$$regret[a] \mathrel{+}= \Pi_{-i} \times (v(a) - v(\sigma^t))$$

$$\sigma[a] = \frac{(regret[a])^+}{\Sigma(regret[i])^+}$$

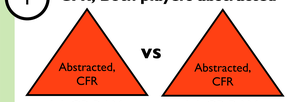
```
}
```

3. As $t \to \infty$, if both players are **regret-minimizing agents**, the **average strategy** over t iterations converges to a Nash equilibrium.

The CFR update rule is just one way to make a regret-minimizing agent. There are other options.

CFR-BR: Use an unabstracted opponent without storing one.

1 CFR, Both players abstracted



Average strategy converges: Optimal in the abstract game, Suboptimal in the real game.

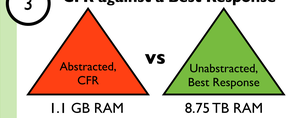
Memory and time efficient.

2 CFR, Opponent is Unabstracted.

Abstracted player's average strategy will converge to minimize exploitability in the real game!

However, high memory requirements make this infeasible.

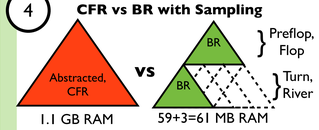
3 CFR against a Best Response



A Best Response is also **regret-minimizing**, so CFR converges. Current strategy converges too!

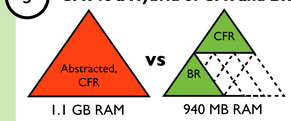
Pure strategies are compact, but still requires too much memory.

4 CFR vs BR with Sampling



Split the game in to a **trunk** and a set of **subgames**. Subgames can be computed as needed - avoids storing everything! But computing trunk strategy requires a full game tree traversal (76 CPU-days!)

5 CFR vs a Hybrid of CFR and BR



Have opponent use CFR in the trunk and BR in the subgames. Now it's time efficient, and uses less memory than the abstracted strategy!
Current strategy isn't guaranteed to converge, but might in practice.

Parallel Implementation
The process can allocate one CFR strategy and one trunk strategy, and use multiple cores to sample subgames.

In practice, this can safely be done without needing to coordinate the threads as collisions only result in the loss of a small update.

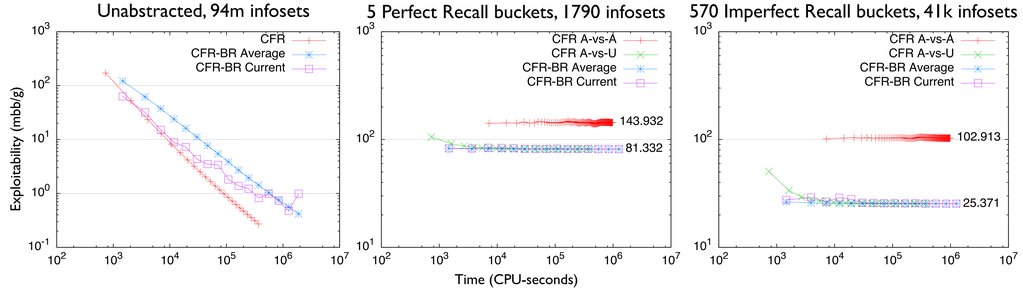
CFR-BR requires almost 1/4 the memory of CFR!
The Hybrid-agent requires very little memory compared to the abstracted CFR agent, and each position can be solved separately. That's 1/2.

Since the current strategy converges in practice, the CFR agent's average strategy is unnecessary.
That's another 1/2.

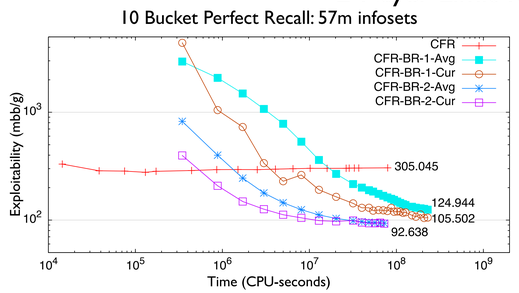
Dividing the Trunk from the Subgames

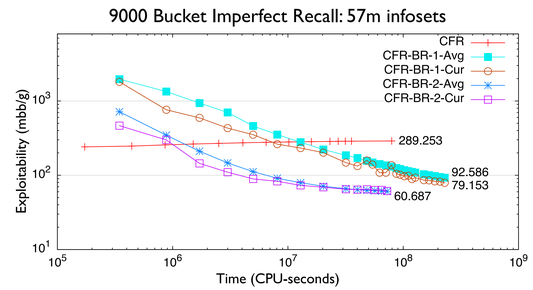
	Heads-Up Limit Hold'em RAM required		
Trunk Rounds	Trunk	One Subgame	Total (48 cores)
1	15 KB	1.18 GB	56.64 GB
2	936 MB	2.74 MB	1.07 GB
3	360 GB	6.54 KB	360 GB
4 (CFR)	140 TB	n/a	140 TB

Verifying in [2-4] Hold'em

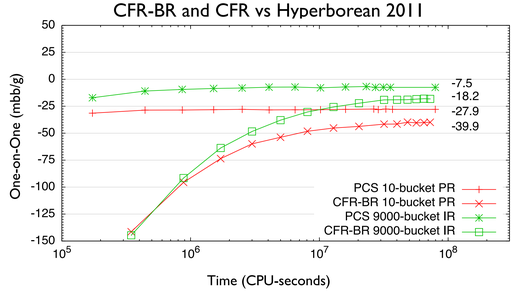


2-Player Limit Texas Hold'em Poker

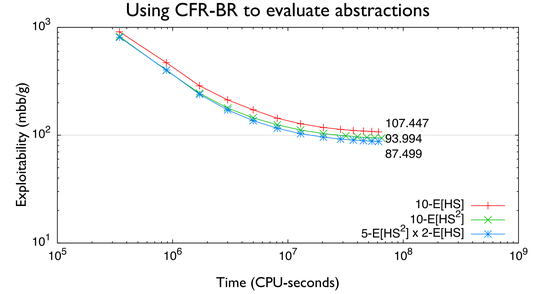




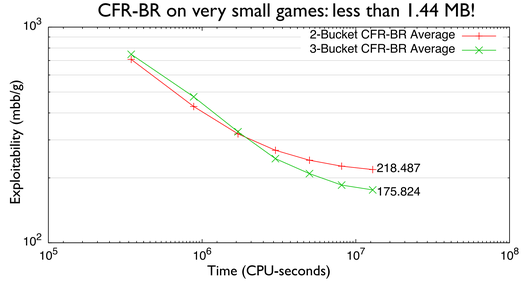
CFR-BR and CFR vs Hyperborean 2011



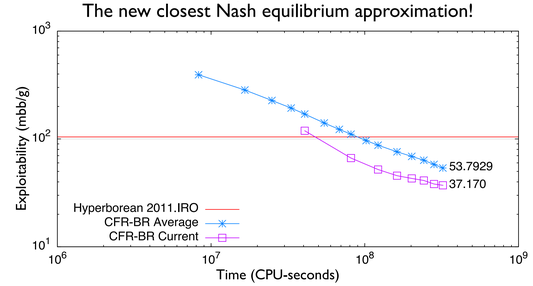
Using CFR-BR to evaluate abstractions



CFR-BR on very small games: less than 1.44 MB!



CFR-BR in the Hyperborean 2011 Abstraction: The new closest Nash equilibrium approximation!



UNIVERSITY OF ALBERTA
Alberta Innovates
compute • calcul CANADA
WestGrid
RQCHP
NSERC CRSNG
Making IT happen
Computing Science