Computing Approximate Nash Equilibria and Robust Best-Responses Using Sampling

Marc Ponsen Steven de Jong Marc Lanctot

Paper Review by Oron Anschel

July 15, 2015

Outline

Outline

Games

Games Examples:

- Puzzles
- Rock-paper-scissors
- Backgammon
- Chess
- Poker
- Video games

Normal Game

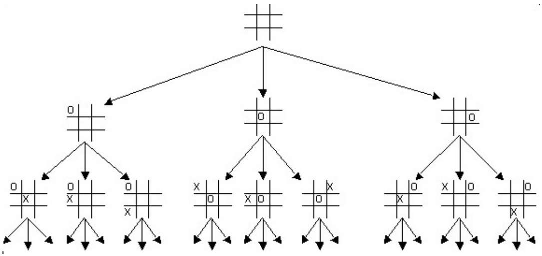
- Players act simultaneously
- Represented in a **Game-Table**
- Example: Rock-paper-scissors

Rock-paper-scissors - Table representation

P1\P2	Rock	Paper	Scissor
Rock	[0,0]	[0,1]	[1,0]
Paper	[1,0]	[0,0]	[0,1]
Scissor	[0,1]	[1,0]	[0,0]

Extensive-Form Game

- Represented as a **Game-Tree**
- Examples : Chess, Backgammon, Poker, Tic Tac Toe
- Characteristics:
  - Sequential decision-making
  - Imperfect information
  - Stochastic



Outline

Best Response Strategy

Assume 2 players game

- σ_i- Player i strategy
- u_i- Player i game utility

Best Response Value

$$b_1(\sigma_2) = \max_{\sigma_1' \in \Sigma_1} u_1(\sigma_1', \sigma_2)$$

Best Response Strategy

$$\sigma_1 = \arg\max_{\sigma_1' \in \Sigma_1} u_1(\sigma_1', \sigma_2)$$

Nash-Equilibrium Strategy

- σ_i- Player i Nash-Equilibrium strategy
- u_i- Player i game utility

Nash-Equilibrium

$$\begin{cases} u_1(\sigma_{1,},\sigma_2) \geq \max\limits_{\sigma_1' \in \Sigma_1} u_1(\sigma_1',\sigma_2) \\ u_2(\sigma_{2,},\sigma_1) \geq \max\limits_{\sigma_2' \in \Sigma_2} u_2(\sigma_2',\sigma_1) \end{cases}$$

Approximate Nash-Equilibrium

$$\begin{cases} u_1(\sigma_{1,},\sigma_2) + \varepsilon \geq \max\limits_{\sigma_1' \in \Sigma_1} u_1(\sigma_1',\sigma_2) \\ u_2(\sigma_{2,},\sigma_1) + \varepsilon \geq \max\limits_{\sigma_2' \in \Sigma_2} u_2(\sigma_2',\sigma_1) \end{cases}$$

*How to compute a Nash-Equilibrium strategy?

Outline

Non-Sampling methods

- **Linear programming** - applied to Poker (Billings et al. 2003)
- **Excessive Gap Technique** - applied to Poker (Hoda et al. 2010, Sandholm 2010)

Outline

Sampling methods

- **Monte Carlo Tree Search (MCTS)** - Based on the UCB algorithm (B. Brügmann 1992, R. Coulom 2006, L. Kocsis and Cs. Szepesvári , S. Gelly 2008).
- **Monte Carlo Counterfactual Regret Minimization (MCCFR)** - Based on the Regret Matching algorithm (Martin Zinkevich 2007, Marc Lanctot 2009)

Monte Carlo Tree Search (MCTS)

MCTS

Monte Carlo Tree Search (MCTS)

- Convergence guarantees for **perfect information** games.

Repeat:

1. Selection:

$$a^* \in \text{argmax}_{a \in A} \left(v_a + C \cdot \sqrt{\frac{\ln n_p}{n_a}} \right)$$

v_a – average simulated reward

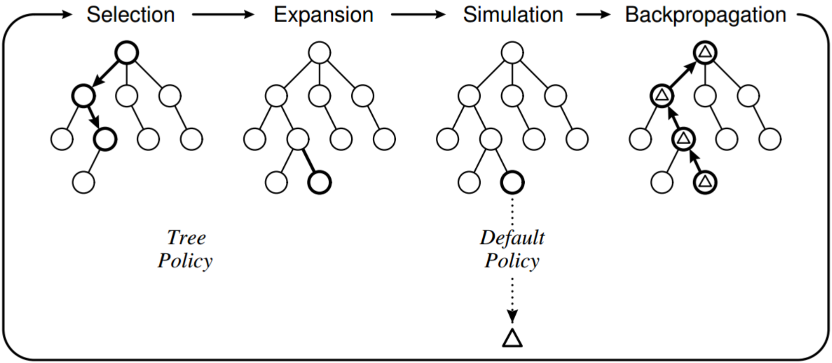
n_a – visit count of action a

n_p – visit counts of current node

(UCB1 algorithm)

2. Expansion
3. Simulation
4. Backpropogation

Monte Carlo Tree Search Cont'd



Monte Carlo Counterfactual Regret Minimization (MCCFR)

MCCFR

Monte Carlo Counterfactual Regret Minimization (MCCFR)

Some general results...

Average overall regret:

$$R_i^T = \frac{1}{T} \max_{\sigma_i' \in \Sigma_i} \sum_{t=1}^{T} \left(u_i(\sigma_i', \sigma_{-i}^t) - u_i(\sigma^t) \right)$$

Average strategy:

$$\bar{\sigma}_i^T(a|I) = \frac{\sum_{t=1}^T \pi_i^{\sigma^t}(I)\, \sigma^t(a|I)}{\sum_{t=1}^T \pi_i^{\sigma^t}(I)}$$

Theorem

In a zero sum game, if $R_i^T \leq \varepsilon$ then $\bar{\sigma}_i^T$ is a 2ε Nash-Equilibrium strategy.

Monte Carlo Counterfactual Regret Minimization (MCCFR)

More results...

Counterfactual value:

$$v_i(\sigma, I) = \sum_{z \in z_I} \pi^{\sigma}_{-i}(z[I])\, \pi^{\sigma}(z[I], z)\, u_i(z)$$

*Z_I - terminal nodes reachable from I, $z[I]$ - prefix of z in I

Intimidate Counterfactual regret :

$$R^T_{i,imm}(a, I) = \frac{1}{T} \sum_{t=1}^{T} \left(v_i\left(\sigma^t_{(I \to a)}, I\right) - v_i\left(\sigma^t, I\right) \right)$$

$$R^T_{i,imm}(I) = \max_{a \in A(I)} R^T_{i,imm}(a, I)$$

Let $x^+ = \max(x, 0)$

Theorem

$$R^T_i \leq \sum_I R^{T,+}_{i,imm}(I)$$

* Using Regret Matching $R^{T,+}_{i,imm}(I)$ can be driven to zero!

Monte Carlo Counterfactual Regret Minimization (MCCFR)

Regret Matching:

$$\sigma_i^t(a|I) = \frac{R_{i,imm}^{T,+}(I,a)}{\sum_a R_{i,imm}^{T,+}(I,a)}$$

- $R_{i,imm}^{T,+}(I,a)$ can be calculated recursively during the tree traversal.
- Can we avoid making full tree traversal?

Monte Carlo Counterfactual Regret Minimization (MCCFR)

Yes!

- MCCFR - Outcome-Sampling.
- Let $\pi^{\sigma'}(z)$ be the probability of sampling z.

Sampled Counterfactual value:

$$\tilde{v}_i(\sigma, I) = \frac{1}{\pi^{\sigma'}(z)} \pi^{\sigma}_{-i}(z[I]) \pi^{\sigma}(z[I], z) u_i(z)$$

- We have that $E[\tilde{v}_i(\sigma, I)] = v_i(\sigma, I)$.
- Sampling based algorithm that convergence to NE.

Outline

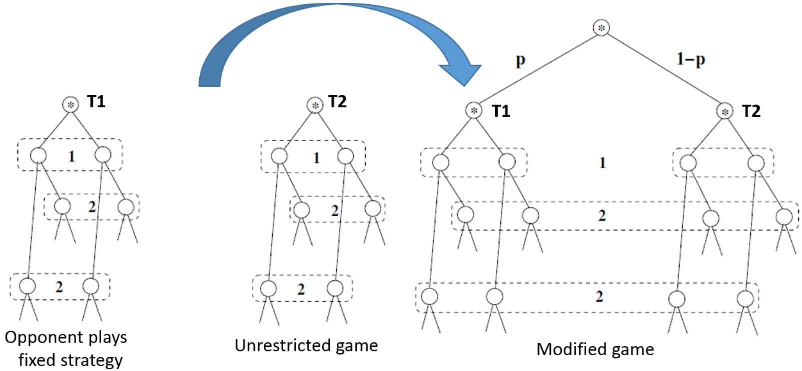
Restricted Nash Response

- What if the opponent doesn't play NES?
- What is the problem in playing best response?
- Can we exploit while being robust?
- **RNR (Johanson et al. 2008)**

Restricted Nash Response Cont'd

What is RNR?

- Robust best response strategy.
- Assume the opponent plays σ_{fix} with probability p .
- Solve a NE for a modified game where the opponent plays $p\sigma_{fix} + (1-p)\sigma_2$.



Outline

Monte Carlo Restricted Nash Response

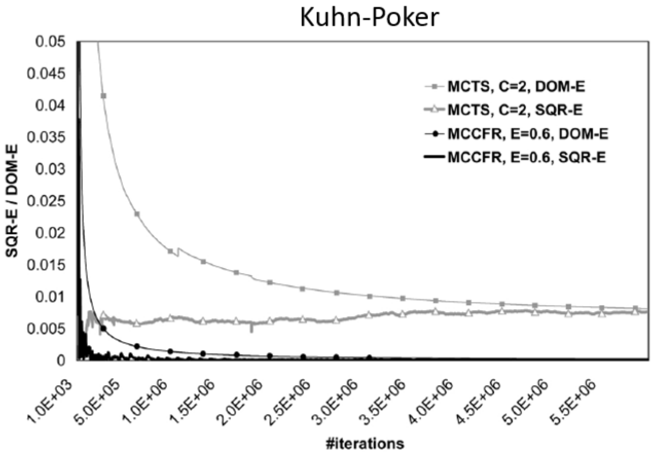
MCRNR Algorithm:

- Evaluate σ_{fix} for the players offline.
- Confidence parameter p can be evaluated for each node/ globally.
- Run MCCFR, use a modified tree as input (do not update fixed strategies nodes).

Outline

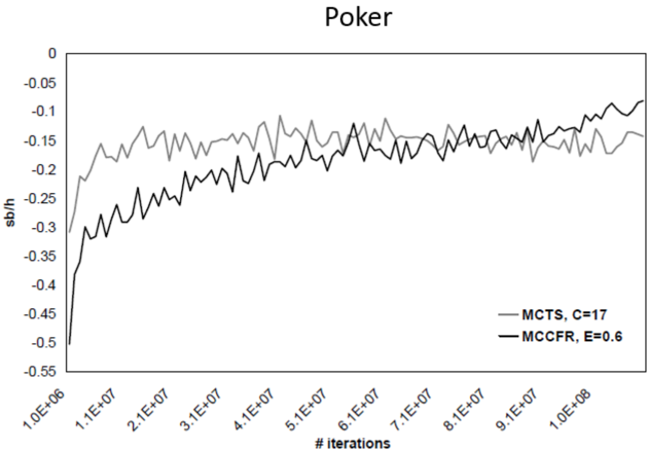
Experiments Results

- MCCFR vs MCTS in Kuhn Poker



Experiments Results Cont'd

- MCCFR vs MCTS in Poker



Experiments Results Cont'd

- Playing against SparBot and POKI (benchmark machine players).
- Each 1000 online games, 5 million MCCFR/MCRNR offline iterations.
- Results obtained after 10,000 online games.

Opponent	MCCFR10	MCRNR10	MCCFR100	MCRNR100
POKI	0.059	0.369	0.191	0.482
SPARBOT	-0.091	-0.039	0.046	0.061

Outline

Contributions

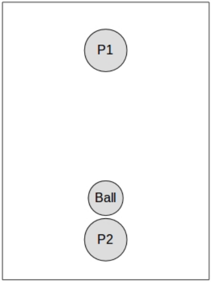
- Comparison between MCTS and MCCFR on two-player Limit Texas Hold'Em Poker.
- Introduced MCRNR algorithm for robust best response strategies.

Outline

Game Setup

Penalty Kick Game:

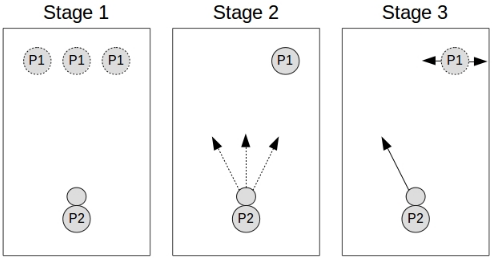
- 2 players and a ball



Game Setup Cont'd

Penalty Kick Game:

- Player 1 : Choose start position
- Player 2 : Choose shot direction
- Player 1 : Move left/right/don't move
- Result : Goal/ no goal



Outline

Results

Nash-Equilibrium Strategy :

- Player 1 : Start at the center
- Player 2 : Choose shot direction (doesn't matter)
- Player 1 : Move to shooting direction
- Result : Player 1 always stops the ball