Computer Poker Tutorial @ EC 2016, part 2: Equilibrium Computation

Marc Lanctot

Google DeepMind

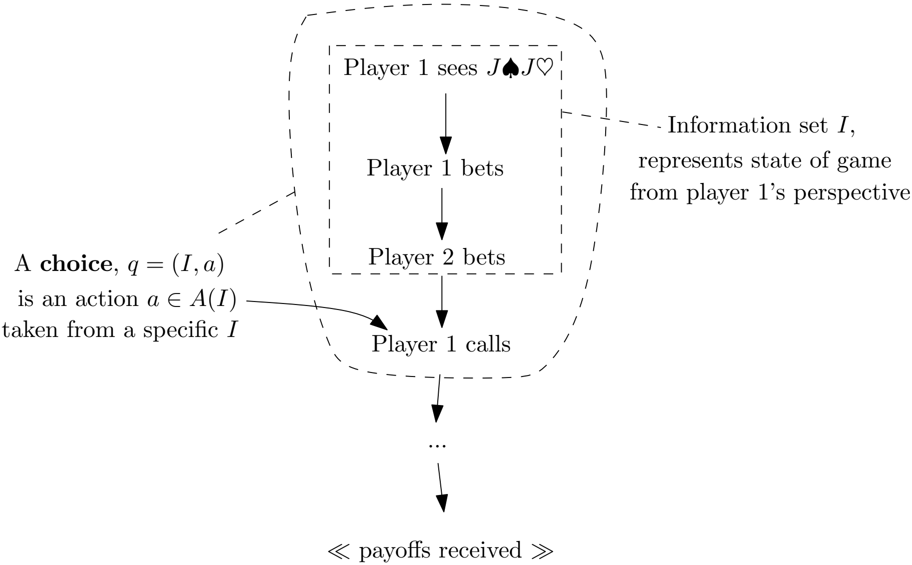
Jul 25th, 2016

Outline

- Optimization using the sequence form
- Counterfactual Regret (CFR) Minimization
- Selected CFR extensions
- Solving 2-player Heads-up Limit Texas Hold'em
- Two open problems

Slides and references available at:
`http://mlanctot.info/ecpokertutorial2016/`

Extensive-form Games



Information sets: $I \in \mathcal{I}$, (context-specific) choices $q = (I, a) \in Q$.

Defaults

Unless otherwise noted, assume:

- Notation based on [Osborne & Rubinstein '94]
- Perfect recall
- Mixed (randomized) strategies are used σ
- Two players: $N = 2$
  - subscript i refers to a player i
  - subscript $-i$ refers to the opponent(s) of player i
- Zero-sum: for every game outcome z, $\sum_{i=1}^{N} u_i(z) = 0$
  - Set of Nash eq. profiles $\{\sigma^*\} \Leftrightarrow$ set of minimax profiles
  - Expected values for eq. $u_i(\sigma^*)$ are unique
  - Nash strategies for player i are interchangeable: $u_i(\sigma^*_{i,1}, \sigma^*_{-i}) = u_i(\sigma^*_{i,2}, \sigma^*_{-i})$

Sequence-Form Representation

Refs: [Koller, Megiddo, von Stengel '94][von Stengel '07]

Let $Q_i = \{(I, a) \mid I \in \mathcal{I}, a \in A(I)\}$ be the set of choices for player i.

Sequence-Form Representation

Refs: [Koller, Megiddo, von Stengel '94][von Stengel '07]

Let $Q_i = \{(I, a) \mid I \in \mathcal{I}, a \in A(I)\}$ be the set of choices for player i.

Encode *realization plan* for player i using constraints; $\Delta Q :=$ all $\mathbf{x}$ such that

- $x_i(q_\emptyset) = 1$
- $x_i(q) = \sum_{q' \in succ_i(q)} x_i(q')$

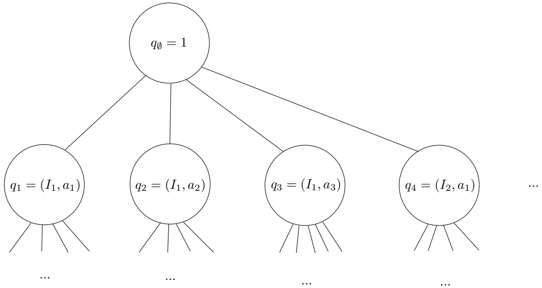
Sequence-Form Representation

Refs: [Koller, Megiddo, von Stengel '94][von Stengel '07]

Let $Q_i = \{(I, a) \mid I \in \mathcal{I}, a \in A(I)\}$ be the set of choices for player i.

Encode *realization plan* for player i using constraints; $\Delta Q :=$ all $\mathbf{x}$ such that

- $x_i(q_\emptyset) = 1$
- $x_i(q) = \sum_{q' \in succ_i(q)} x_i(q')$



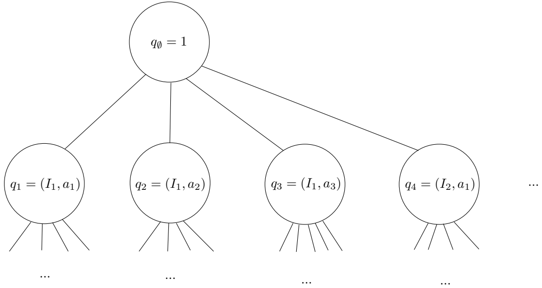
Sequence-Form Representation

Refs: [Koller, Megiddo, von Stengel '94][von Stengel '07]

Let $Q_i = \{(I,a) \mid I \in \mathcal{I}, a \in A(I)\}$ be the set of choices for player i.

Encode *realization plan* for player i using constraints; $\Delta Q :=$ all $\mathbf{x}$ such that

- $x_i(q_\emptyset) = 1$
- $x_i(q) = \sum_{q' \in succ_i(q)} x_i(q')$



$\Rightarrow$ Much more space-efficient than mixture over pure strategies!

Sequence-Form Linear Programming

For two-player zero-sum, setup an optimization problem:

$$\max_{\mathbf{x}\in\Delta Q_1}\min_{\mathbf{y}\in\Delta Q_2}\mathbf{xAy}=\min_{\mathbf{y}\in\Delta Q_2}\max_{\mathbf{x}\in\Delta Q_1}\mathbf{xAy},$$

Subject to $\mathbf{Ex}=\mathbf{e},\mathbf{x}\geq\mathbf{0},\mathbf{x}\cdot\mathbf{1}=1$.

Sequence-Form Linear Programming

For two-player zero-sum, setup an optimization problem:

$$\max_{\mathbf{x}\in\Delta Q_1} \min_{\mathbf{y}\in\Delta Q_2} \mathbf{xAy} = \min_{\mathbf{y}\in\Delta Q_2} \max_{\mathbf{x}\in\Delta Q_1} \mathbf{xAy},$$

Subject to $\mathbf{Ex} = \mathbf{e}, \mathbf{x} \geq \mathbf{0}, \mathbf{x} \cdot \mathbf{1} = 1$.

Here:

- $\mathbf{A}$ has an entry for every q_1 and q_2 that result in terminal states.
- $\mathbf{E}$ encodes the structure of Q_i
- $\mathbf{e}$ is $(1, 0, 0, 0, ...)^T$

Sequence-Form Linear Programming

For two-player zero-sum, setup an optimization problem:

$$\max_{\mathbf{x}\in\Delta Q_1}\min_{\mathbf{y}\in\Delta Q_2}\mathbf{xAy}=\min_{\mathbf{y}\in\Delta Q_2}\max_{\mathbf{x}\in\Delta Q_1}\mathbf{xAy},$$

Subject to $\mathbf{Ex}=\mathbf{e},\mathbf{x}\geq\mathbf{0},\mathbf{x}\cdot\mathbf{1}=1$.

Here:

- $\mathbf{A}$ has an entry for every q_1 and q_2 that result in terminal states.
- $\mathbf{E}$ encodes the structure of Q_i
- $\mathbf{e}$ is $(1,0,0,0,...)^T$

Storing $\mathbf{A}$ requires $O(|Q_1||Q_2|)$ space in the worst case.

Sequence-Form Linear Programming

For two-player zero-sum, setup an optimization problem:

$$\max_{\mathbf{x}\in\Delta Q_1} \min_{\mathbf{y}\in\Delta Q_2} \mathbf{xAy} = \min_{\mathbf{y}\in\Delta Q_2} \max_{\mathbf{x}\in\Delta Q_1} \mathbf{xAy},$$

Subject to $\mathbf{Ex} = \mathbf{e}, \mathbf{x} \geq \mathbf{0}, \mathbf{x} \cdot \mathbf{1} = 1$.

Here:

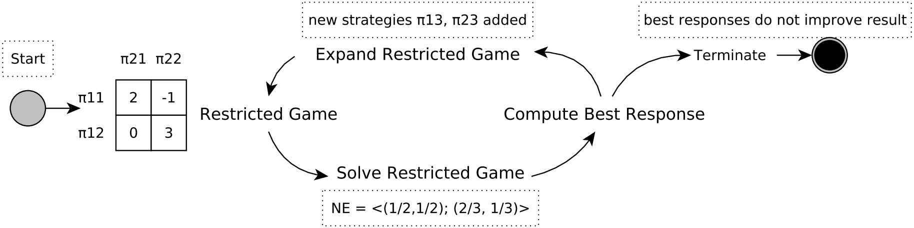
- $\mathbf{A}$ has an entry for every q_1 and q_2 that result in terminal states.
- $\mathbf{E}$ encodes the structure of Q_i
- $\mathbf{e}$ is $(1, 0, 0, 0, ...)^T$

Storing $\mathbf{A}$ requires $O(|Q_1||Q_2|)$ space in the worst case.

For large games, in practice, $\mathbf{A}$ is sparse and so memory requirements are closer to $|Q_1| + |Q_2|$.

Double-Oracle Methods

Refs: [Zinkevich et al. '06][Bosansky et. al '14]



In extensive-form:

- Each row/column corresponds to a realization plan
- Heuristics to compute "good" best responses
- Could still require enumerating entire space

Nesterov's Excessive Gap Technique (EGT)

Refs: [Gilpin et al. '07][Hoda et al. '10]

Recall:

$$\max_{\mathbf{x}\in\Delta Q_1} f(\mathbf{y}) = \min_{\mathbf{y}\in\Delta Q_2} \phi(\mathbf{x}),$$

where $f(\mathbf{y}) = \min_{\mathbf{y}\in\Delta Q_2} z, \quad \phi(\mathbf{x}) = \max_{\mathbf{x}\in\Delta Q_1} z, \quad z = \mathbf{xAy}.$

Use a strongly convex function d_i on ΔQ_i, and define:

$$f_{\mu_2}(\mathbf{y}) = \min_{\mathbf{y}\in Q_2}\{z + \mu_2 d_2(\mathbf{y})\}$$

$$\phi_{\mu_1}(\mathbf{x}) = \max_{\mathbf{x}\in Q_1}\{z - \mu_1 d_1(\mathbf{x})\}$$

Then $\quad f_{\mu_2}(\mathbf{y}) \geq \phi_{\mu_1}(\mathbf{x}) \quad \Rightarrow \quad 0 \leq \phi(\mathbf{y}) - f(\mathbf{x}) \leq \mu_1 d_1^\top + \mu_2 d_2^\top$, so iteratively compute $(\mathbf{x}^k, \mathbf{y}^k, \mu_1^k, \mu_2^k)$ such that $\mu_i^{k+1} < \mu_i^k$ by gradient descent.

Theorem: EGT computes an ϵ-equlibrium in $O(1/\epsilon)$ iterations.

Counterfactual Regret Minimization (CFR)

CFR [Zinkevich et al. 2008] is iterative strategy-updating algorithm:

	$t = 1$
Player 1 strategies:	σ_1^1
Player 2 strategies:	σ_2^1

Counterfactual Regret Minimization (CFR)

CFR [Zinkevich et al. 2008] is iterative strategy-updating algorithm:

$$
\begin{array}{lccc}
 & t = 1 & & t = 2 \\
\text{Player 1 strategies:} & \sigma_1^1 & \rightarrow & \sigma_1^2 \\
\text{Player 2 strategies:} & \sigma_2^1 & \rightarrow & \sigma_2^2
\end{array}
$$

Counterfactual Regret Minimization (CFR)

CFR [Zinkevich et al. 2008] is iterative strategy-updating algorithm:

$$\begin{array}{lccccccc} & t=1 & & t=2 & & t=3 & \cdots \\ \text{Player 1 strategies:} & \sigma_1^1 & \rightarrow & \sigma_1^2 & \rightarrow & \sigma_1^3 & \cdots \\ \text{Player 2 strategies:} & \sigma_2^1 & \rightarrow & \sigma_2^2 & \rightarrow & \sigma_2^3 & \cdots \end{array}$$

Counterfactual Regret Minimization (CFR)

CFR [Zinkevich et al. 2008] is iterative strategy-updating algorithm:

$$\begin{array}{lccccccc} & t=1 & & t=2 & & t=3 & \cdots \\ \text{Player 1 strategies:} & \sigma_1^1 & \to & \sigma_1^2 & \to & \sigma_1^3 & \cdots \\ \text{Player 2 strategies:} & \sigma_2^1 & \to & \sigma_2^2 & \to & \sigma_2^3 & \cdots \end{array}$$

Let R_i^T be the **external regret** of using σ^t after T steps:

$$R_i^T = \max_{a \in A} \mathbb{E}\left[\sum_{t=1}^{T} \left(u_i(a, \sigma_{-i}^t) - u_i(\sigma_i^t, \sigma_{-i}^t)\right)\right]$$

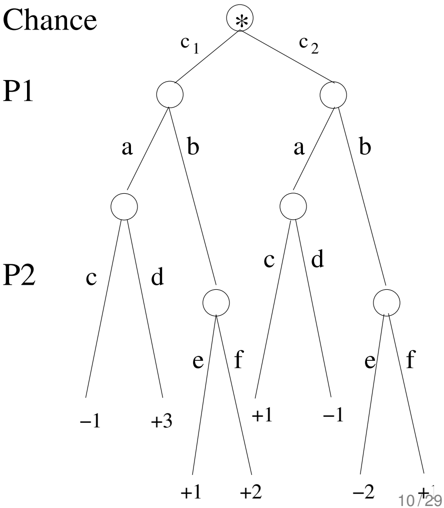
$R_i^T / T \leq \epsilon \quad \Rightarrow \quad$ the average profile $(\bar{\sigma}_1^T, \bar{\sigma}_2^T)$ is a 2ϵ-Nash.

- σ is ϵ-Nash if a player can do ϵ better by switching to σ_i'.
- σ is Nash if no player can do better by switching strategies.

Terminology I : Extensive-Form Games

Example:

An **extensive-form game** is represented in tree form.

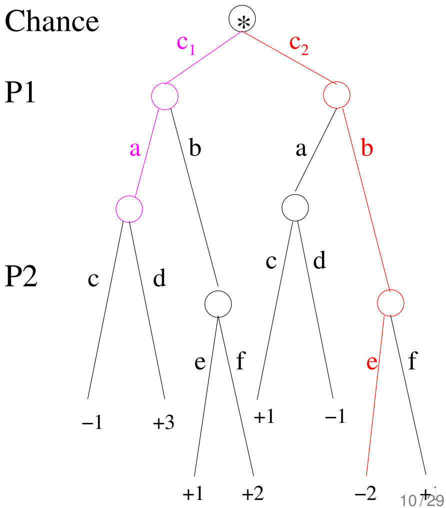


Terminology I : Extensive-Form Games

An **extensive-form game** is represented in tree form.

- $h \in H$ is possible history; $z \in Z, Z \subseteq H$ is a terminal history.

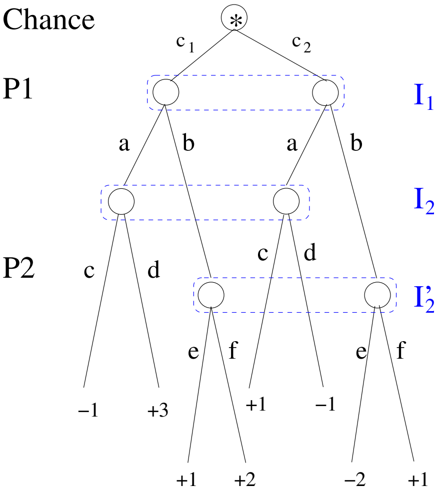
Example:



Terminology I : Extensive-Form Games

Example:

An **extensive-form game** is represented in tree form.



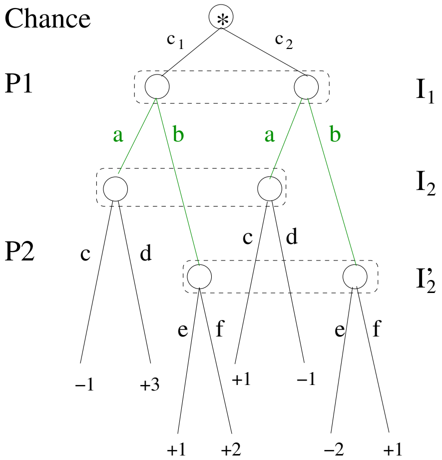
- $h \in H$ is possible history; $z \in Z, Z \subseteq H$ is a terminal history.
- An information set $I_i \in \mathcal{I}$ is an information set for player i.

Terminology I : Extensive-Form Games

An **extensive-form game** is represented in tree form.

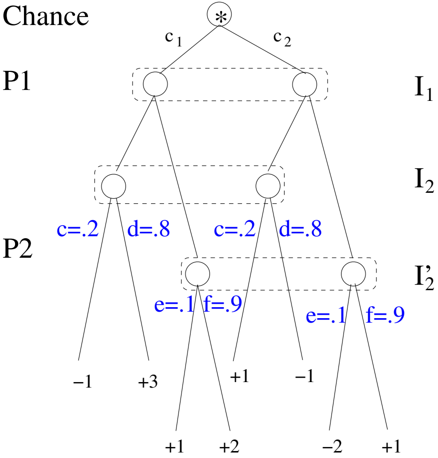
- $h \in H$ is possible history;
  $z \in Z, Z \subseteq H$ is a terminal history.
- An information set $I_i \in \mathcal{I}$ is an information set for player i.
- $A(I_i)$ is the action set for i at information set I_i.

Example:



Terminology II : Strategies

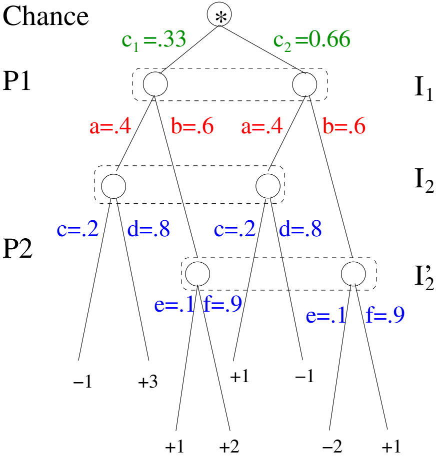
A **strategy** $\sigma_i \in \Sigma_i$ is a distribution from $I_i \rightarrow A(I_i)$.



Terminology II : Strategies

A **strategy** $\sigma_i \in \Sigma_i$ is a distribution from $I_i \rightarrow A(I_i)$.

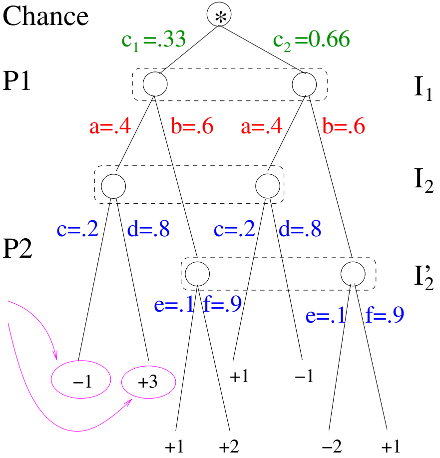
- A strategy σ_{-i} is a strategy for the opponents of i and chance.
- A strategy profile $\sigma = (\sigma_1, \sigma_2)$.



Terminology II : Strategies

A **strategy** $\sigma_i \in \Sigma_i$ is a distribution from $I_i \to A(I_i)$.

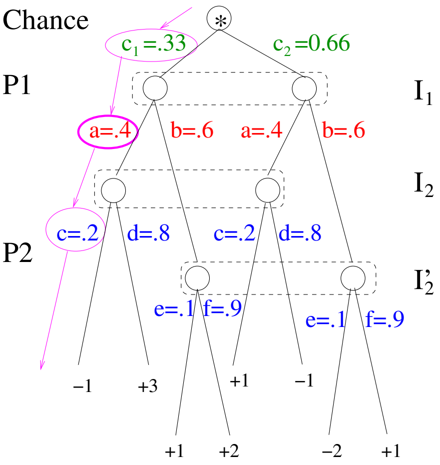
- A strategy σ_{-i} is a strategy for the opponents of i and chance.
- A strategy profile $\sigma = (\sigma_1, \sigma_2)$.
- $u_i(z)$ is the payoff to player i when players play z.



Terminology II : Strategies

A **strategy** $\sigma_i \in \Sigma_i$ is a distribution from $I_i \rightarrow A(I_i)$.

- A strategy σ_{-i} is a strategy for the opponents of i and chance.
- A strategy profile $\sigma = (\sigma_1, \sigma_2)$.
- $u_i(z)$ is the payoff to player i when players play z.
- $\pi^\sigma(h)$ is a product of probabilities along history h. $\pi_i^\sigma(h)$ is player i's contribution.



CFR Algorithm (Overview)

Refs: [Zinkevich et al. '08][Hart & Mas-Colell '00]

1. Minimize **average immediate counterfactual regret** $R^T_{i,\text{imm}}(I)$
2. **Theorem 3**: Overall regret bounded by

$$R^T_i/T \leq \sum_{I \in \mathcal{I}_i} R^{T,+}_{i,\text{imm}}(I)$$

3. **Theorem 4**: Using regret-matching to update strategies, σ^t at each information set, then

$$R^T_i/T \leq \frac{\Delta_{u,i}|\mathcal{I}_i|\sqrt{|A_i|}}{\sqrt{T}}$$

where $\Delta_{u,i}$ is a payoff range for i.

CFR Algorithm (Example)

Define **counterfactual value** as

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

Define $v_i(\sigma_{(I \to a)}, I)$ similarly, except take a at I

CFR Algorithm (Example)

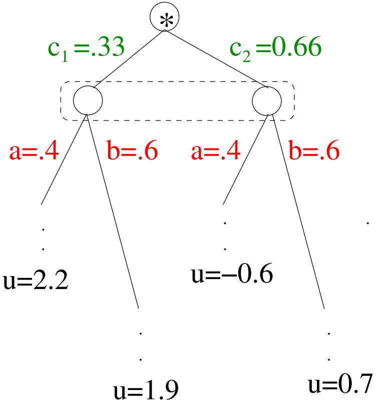
Define **counterfactual value** as

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

Define $v_i(\sigma_{(I \to a)}, I)$ similarly, except take a at I

Repeat until sufficiently small ϵ:

1. Walk the game tree computing $r(I, a) = v_i(\sigma_{(I \to a)}, I) - v_i(\sigma, I)$



CFR Algorithm (Example)

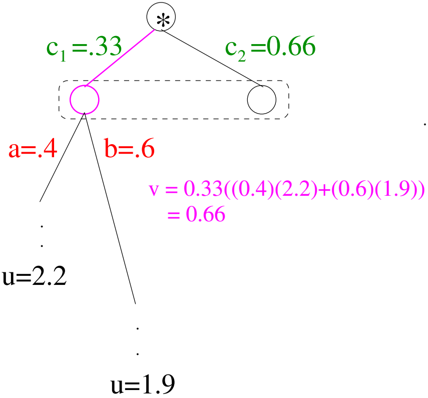
Define **counterfactual value** as

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

Define $v_i(\sigma_{(I \to a)}, I)$ similarly, except take a at I

Repeat until sufficiently small ϵ:

1. Walk the game tree computing $r(I, a) = v_i(\sigma_{(I \to a)}, I) - v_i(\sigma, I)$
   1. Recursively compute $r(I, a)$ at a particular node
   2. Add to accumulated values $r[I, a] \mathrel{+}= r(I, a)$



CFR Algorithm (Example)

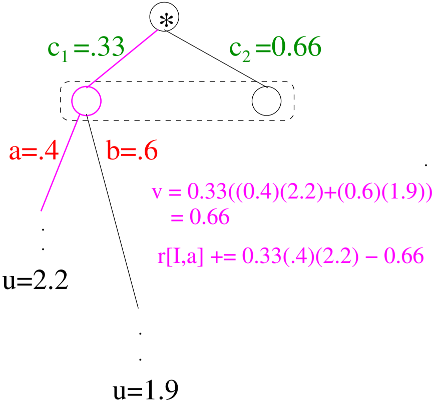
Define **counterfactual value** as

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

Define $v_i(\sigma_{(I \to a)}, I)$ similarly, except take a at I

Repeat until sufficiently small ϵ:

1. Walk the game tree computing $r(I,a) = v_i(\sigma_{(I \to a)}, I) - v_i(\sigma, I)$
   1. Recursively compute $r(I,a)$ at a particular node
   2. Add to accumulated values $r[I,a] \mathrel{+}= r(I,a)$



CFR Algorithm (Example)

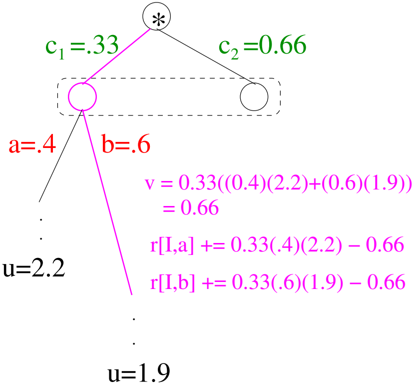
Define **counterfactual value** as

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

Define $v_i(\sigma_{(I \to a)}, I)$ similarly, except take a at I

Repeat until sufficiently small ϵ:

1. Walk the game tree computing $r(I, a) = v_i(\sigma_{(I \to a)}, I) - v_i(\sigma, I)$
   1. Recursively compute $r(I, a)$ at a particular node
   2. Add to accumulated values $r[I, a] \mathrel{+}= r(I, a)$



CFR Algorithm (Example)

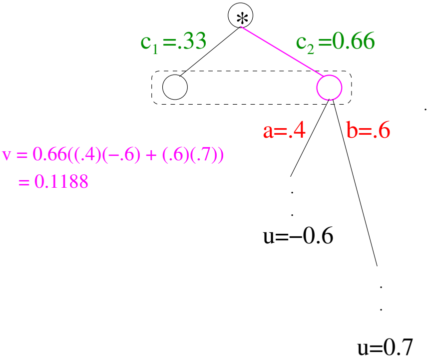
Define **counterfactual value** as

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

Define $v_i(\sigma_{(I \to a)}, I)$ similarly, except take a at I

Repeat until sufficiently small ϵ:

1. Walk the game tree computing $r(I,a) = v_i(\sigma_{(I \to a)}, I) - v_i(\sigma, I)$
   1. Recursively compute $r(I,a)$ at a particular node
   2. Add to accumulated values $r[I,a] \mathrel{+}= r(I,a)$



CFR Algorithm (Example)

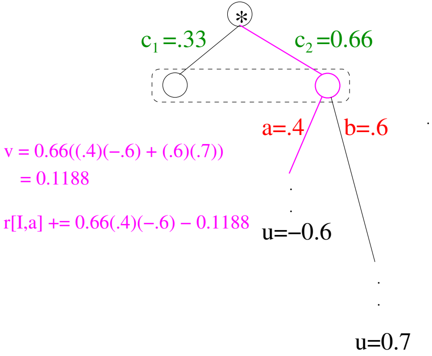
Define **counterfactual value** as

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

Define $v_i(\sigma_{(I\to a)}, I)$ similarly, except take a at I

<u>Repeat until sufficiently small ϵ:</u>

1. Walk the game tree computing $r(I,a) = v_i(\sigma_{(I\to a)}, I) - v_i(\sigma, I)$
    1. Recursively compute $r(I,a)$ at a particular node
    2. Add to accumulated values $r[I,a] \mathrel{+}= r(I,a)$



CFR Algorithm (Example)

Define **counterfactual value** as

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

Define $v_i(\sigma_{(I \to a)}, I)$ similarly, except take a at I

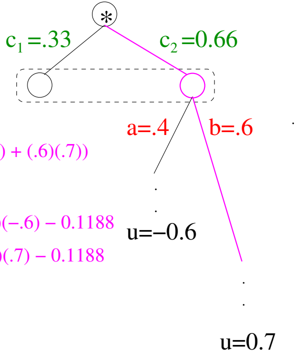
Repeat until sufficiently small ϵ:

1. Walk the game tree computing $r(I, a) = v_i(\sigma_{(I \to a)}, I) - v_i(\sigma, I)$
   1. Recursively compute $r(I, a)$ at a particular node
   2. Add to accumulated values $r[I, a] \mathrel{+}= r(I, a)$

$v = 0.66((.4)(-.6) + (.6)(.7))$
$= 0.1188$

r[I,a] += 0.66(.4)(–.6) – 0.1188

r[I,b] += 0.66(.6)(.7) – 0.1188



CFR Algorithm (Example)

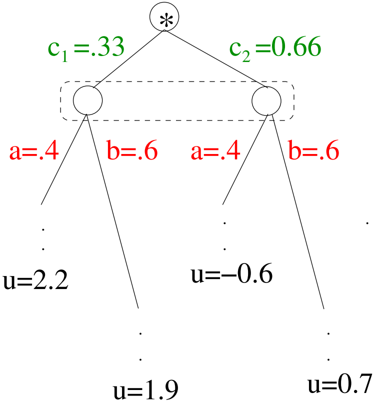
Define **counterfactual value** as

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

Define $v_i(\sigma_{(I \to a)}, I)$ similarly, except take a at I

Repeat until sufficiently small ϵ:

1. Walk the game tree computing $r(I, a) = v_i(\sigma_{(I \to a)}, I) - v_i(\sigma, I)$
    1. Recursively compute $r(I, a)$ at a particular node
    2. Add to accumulated values $r[I, a] \mathrel{+}= r(I, a)$
2. $\sigma_i^{t+1}(I) \leftarrow \text{RegretMatching}(r[I])$



CFR Algorithm (Example)

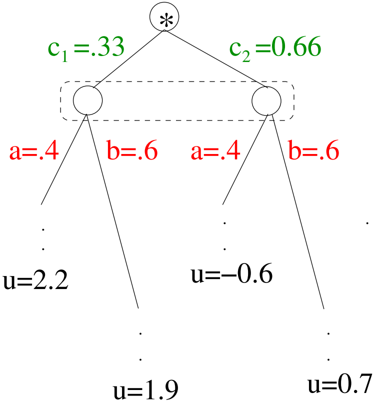
Define **counterfactual value** as

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

Define $v_i(\sigma_{(I \to a)}, I)$ similarly, except take a at I

Repeat until sufficiently small ϵ:

1. Walk the game tree computing $r(I, a) = v_i(\sigma_{(I \to a)}, I) - v_i(\sigma, I)$
    1. Recursively compute $r(I, a)$ at a particular node
    2. Add to accumulated values $r[I, a] \mathrel{+}= r(I, a)$
2. $\sigma_i^{t+1}(I) \leftarrow \text{RegretMatching}(r[I])$
3. Update average profile $\bar{\sigma}$



CFR Extension Outline

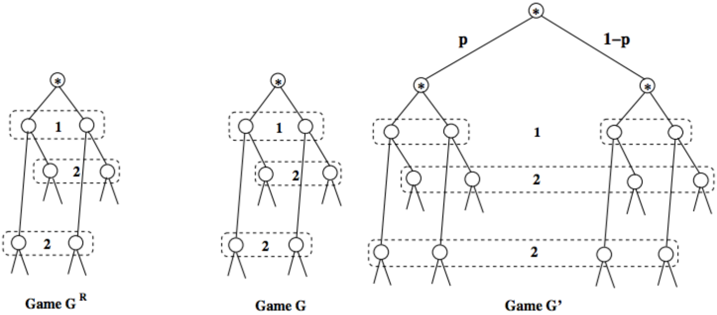
Many (20+ !) follow-up papers on CFR.

I will cover a subset:

- Restricted Nash Responses
- Monte Carlo CFR
- Imperfect Recall Abstraction
- Multiplayer and non-zero-sum
- Sequence-Form Replicator Dynamics
- CFR-BR
- CFR+

Restricted Nash Responses

Refs: [Johanson and Bowling '08, '09][Ponsen et al. '12]



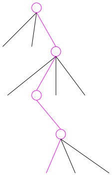
- Game $\mathbf{G}$ is some game.
- $\mathbf{G}^{\mathbf{R}}$ is a *restricted copy* (e.g. player $-i$ plays σ_{fixed})
- Nash$_i(\mathbf{G}')$ $\Leftrightarrow$ best trade-off between Nash$_i(\mathbf{G})$ and BR$_i(\sigma_{fixed})$

Monte Carlo CFR

Refs: [Lanctot et. al '09], [Gibson et al. '12], [Johanson et al. '12], [Burch et al. '12]

Sample parts of the tree: **sampled counterfactual values** $\tilde{v}_i(\sigma, I)$.

Unbiased estimator: $\mathbb{E}[\tilde{v}_i(\sigma, I)] = v_i(\sigma, I)$.



Outcome sampling

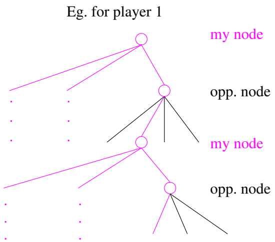
Theorem: with probability $1-p$, δ is i's min prob sampling z

$$R_i^T/T \leq \left(M_i(\sigma_i^*) \sqrt{|\max_I A(I)|} + \frac{\sqrt{2|\mathcal{I}_i||\mathcal{B}_i|}}{\sqrt{p}} \right) \left(\frac{1}{\delta} \right) \left(\frac{\Delta_{u,i}}{\sqrt{T}} \right)$$

E.g. chance sampling $\rightarrow$ sample only chance outcomes

Monte Carlo CFR: External Sampling

Refs: [Lanctot et. al ’09][Gibson ’14]



Theorem: with prob $1-p$:

$$R_i^T/T \leq \left(M_i(\sigma_i^*)\sqrt{|\max_I A(I)|} + \frac{\sqrt{2|\mathcal{I}_i||\mathcal{B}_i|}}{\sqrt{p}} \right) \left(\frac{\Delta_{u,i}}{\sqrt{T}} \right)$$

- Has worked well in (>2)-player and large action spaces
- Tartanian7, 2014 winner of 2P NL, used variant of ext. sampling

Monte Carlo CFR: Public Chance Sampling

Refs: [Johanson et. al '11][Jackson '12]

Sample only *public* chance events!

Monte Carlo CFR: Public Chance Sampling

Refs: [Johanson et. al ’11][Jackson ’12]

Sample only *public* chance events!

Vectorize the tree walk (one element per opponent private card)

Monte Carlo CFR: Public Chance Sampling

Refs: [Johanson et. al '11][Jackson '12]

Sample only *public* chance events!

Vectorize the tree walk (one element per opponent private card)

Same bound as E.S. but can use equiv. classes at leaf nodes!

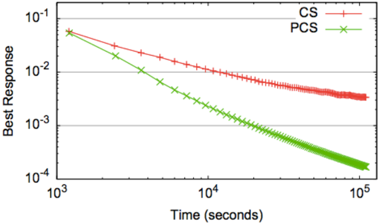
Monte Carlo CFR: Public Chance Sampling

Refs: [Johanson et. al '11][Jackson '12]

Sample only *public* chance events!

Vectorize the tree walk (one element per opponent private card)

Same bound as E.S. but can use equiv. classes at leaf nodes!



Liar's Dice (2,2)

Slumbot, 2012 winner of HULHE, used PCS

Generalized Monte Carlo CFR

Refs: [Gibson et al. '12]

Given *any* estimator for counterfactual values $\hat{v}(I, a)$ with bounded range $\hat{\Delta}$:

Theorem: with prob $1 - p$,

$$R_i^T / T \leq |\mathcal{I}_i| \left(\frac{\hat{\Delta}_i \sqrt{\max_I |A(I)|}}{\sqrt{T}} + \sqrt{\frac{\mathbf{Var}}{pT} + \frac{\mathbf{Cov}}{p} + \frac{\mathbf{E}^2}{p}} \right),$$

where:

- $\mathbf{Var}$ is max variance of diff in regret and est. regret at t,
- $\mathbf{Cov}$ is max covariance of diff in regret and est. regret at t, t',
- $\mathbf{E}$ is the max expectation of diff in regret and est. regret (bias) at t,

over all time steps t (and t'), info sets I, actions $a \in A(I)$.

Imperfect Recall Abstraction

Refs: [Waugh et al. '09][Lanctot et al. '12][Kroer & Sandholm '14, '16]

History $h \in \breve{I}$, define $X_i(h) = (\breve{I}_1, a_1), (\breve{I}_2, a_2), \cdots$ as player i's choice sequence for all I_k belonging to i in h.

Imperfect Recall Abstraction

Refs: [Waugh et al. '09][Lanctot et al. '12][Kroer & Sandholm '14, '16]

History $h \in \breve{I}$, define $X_i(h) = (\breve{I}_1, a_1), (\breve{I}_2, a_2), \cdots$ as player i's choice sequence for all I_k belonging to i in h.

Perfect recall: for all $h, h' \in \breve{I} \Leftrightarrow X_i(h) = X(h')$

Imperfect Recall Abstraction

Refs: [Waugh et al. '09][Lanctot et al. '12][Kroer & Sandholm '14, '16]

History $h \in \breve{I}$, define $X_i(h) = (\breve{I}_1, a_1), (\breve{I}_2, a_2), \cdots$ as player i's choice sequence for all I_k belonging to i in h.

Perfect recall: for all $h, h' \in \breve{I} \Leftrightarrow X_i(h) = X(h')$

Not every $h \in \breve{I}$ is necessarily relevant for computing approx. $\sigma(\breve{I})$!!

Imperfect Recall Abstraction

Refs: [Waugh et al. '09][Lanctot et al. '12][Kroer & Sandholm '14, '16]

History $h \in \breve{I}$, define $X_i(h) = (\breve{I}_1, a_1), (\breve{I}_2, a_2), \cdots$ as player i's choice sequence for all I_k belonging to i in h.

Perfect recall: for all $h, h' \in \breve{I} \Leftrightarrow X_i(h) = X(h')$

Not every $h \in \breve{I}$ is necessarily relevant for computing approx. $\sigma(\breve{I})$!!

Purposely *forget* parts of $h \in \breve{I}$ and $h' \in \breve{I}'$; $\rightarrow$ merge $I = \breve{I} \cup \breve{I}'$.

Imperfect Recall Abstraction

Refs: [Waugh et al. '09][Lanctot et al. '12][Kroer & Sandholm '14, '16]

History $h \in \breve{I}$, define $X_i(h) = (\breve{I}_1, a_1), (\breve{I}_2, a_2), \cdots$ as player i's choice sequence for all I_k belonging to i in h.

Perfect recall: for all $h, h' \in \breve{I} \Leftrightarrow X_i(h) = X(h')$

Not every $h \in \breve{I}$ is necessarily relevant for computing approx. $\sigma(\breve{I})$!!

Purposely *forget* parts of $h \in \breve{I}$ and $h' \in \breve{I}'$; $\rightarrow$ merge $I = \breve{I} \cup \breve{I}'$.
Benefits:

1. Huge savings in memory

Imperfect Recall Abstraction

Refs: [Waugh et al. '09][Lanctot et al. '12][Kroer & Sandholm '14, '16]

History $h \in \breve{I}$, define $X_i(h) = (\breve{I}_1, a_1), (\breve{I}_2, a_2), \cdots$ as player i's choice sequence for all I_k belonging to i in h.

Perfect recall: for all $h, h' \in \breve{I} \Leftrightarrow X_i(h) = X(h')$

Not every $h \in \breve{I}$ is necessarily relevant for computing approx. $\sigma(\breve{I})$!!

Purposely *forget* parts of $h \in \breve{I}$ and $h' \in \breve{I}'$; $\rightarrow$ merge $I = \breve{I} \cup \breve{I}'$.
Benefits:

1. Huge savings in memory
2. Often clear what should be forgotten

Imperfect Recall Abstraction

Refs: [Waugh et al. '09][Lanctot et al. '12][Kroer & Sandholm '14, '16]

History $h \in \breve{I}$, define $X_i(h) = (\breve{I}_1, a_1), (\breve{I}_2, a_2), \cdots$ as player i's choice sequence for all I_k belonging to i in h.

Perfect recall: for all $h, h' \in \breve{I} \Leftrightarrow X_i(h) = X(h')$

Not every $h \in \breve{I}$ is necessarily relevant for computing approx. $\sigma(\breve{I})$!!

Purposely *forget* parts of $h \in \breve{I}$ and $h' \in \breve{I}'$; $\rightarrow$ merge $I = \breve{I} \cup \breve{I}'$.
Benefits:

1. Huge savings in memory
2. Often clear what should be forgotten
3. CFR algorithm still runs(!)
   - But does it still work/converge?
   - In theory: yes! Under some (somewhat restrictive) assumptions.
   - In practice: yes, very well!

Multi (> 2) player and non-zero sum

Refs: [Abou Risk & Szafron '10][Gibson & Szafron '11][Gibson et al. '13][Gibson '14]

Generally not much known about CFR in this case.

But here again, algorithm is still well-defined.

Gibson 2014:

- Regret min. removes iteratively strictly-dominated strategies.
- Extend to *dominated actions* and counterfactual values.
- CFR removes iterative strictly-dominated actions.
- 2-player game: If $R_i^T/T < \epsilon$, converges to $2(\epsilon + \delta_u)$-Nash.

Hyperborean: winner of 2012, 2013, and 2014 3-player competitions.

Sequence-Form Replicator Dynamics

Refs: [Gatti et al. '13][Lanctot '14]

Recall Q set of choices (I, a), and $x_i(q)$ realization weight on q:

For player i, for each $q \in Q_i$, update:

$$x_i(q, t+1) = x_i(q, t)\frac{u_i(\mathbf{x}_{i \to g_q})}{u_i(\mathbf{x})}$$

Sequence-Form Replicator Dynamics

Refs: [Gatti et al. '13][Lanctot '14]

Recall Q set of choices (I, a), and $x_i(q)$ realization weight on q:

For player i, for each $q \in Q_i$, update:

$$x_i(q, t+1) = x_i(q, t)\frac{u_i(\mathbf{x}_{i \to g_q})}{u_i(\mathbf{x})}$$

$\mathbf{x}_{i \to g_q}$ is $\mathbf{x}$
except player i
uses $g_q(\mathbf{x}_i)$

Sequence-Form Replicator Dynamics

Refs: [Gatti et al. '13][Lanctot '14]

Recall Q set of choices (I, a), and $x_i(q)$ realization weight on q:

For player i, for each $q \in Q_i$, update:

$$x_i(q, t+1) = x_i(q, t) \frac{u_i(\mathbf{x}_{i \to g_q})}{u_i(\mathbf{x})}$$

$\mathbf{x}_{i \to g_q}$ is $\mathbf{x}$
except player i
uses $\underline{g_q(\mathbf{x}_i)}$

$$g_q(\mathbf{x}_i, q') = \begin{cases} 1 & \text{if } q' \in X_i(q), \\ \frac{x_i(q')}{\text{Ancestor}(q, q')} & \text{if } X_i(q) \sqsubseteq X_i(q'), \\ 0 & \text{otherwise,} \end{cases}$$

Sequence-Form Replicator Dynamics

Refs: [Gatti et al. '13][Lanctot '14]

For player i, for each $q \in Q_i$, update:

$$x_i(q, t+1) = x_i(q, t)\frac{u_i(\mathbf{x}_{i\to g_q})}{u_i(\mathbf{x})}$$

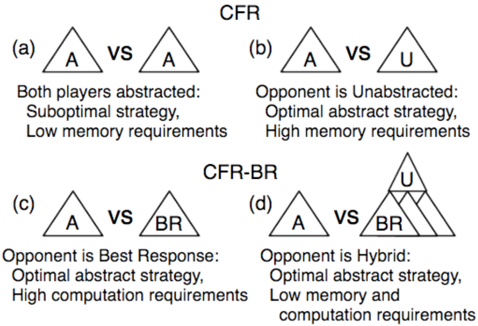
$\mathbf{x}_{i\to g_q}$ is $\mathbf{x}$
except player i
uses $g_q(\mathbf{x}_i)$

$$g_q(\mathbf{x}_i, q') = \begin{cases} 1 & \text{if } q' \in X_i(q), \\ \frac{x_i(q')}{\text{Ancestor}_{(q,q')}} & \text{if } X_i(q) \sqsubseteq X_i(q'), \\ 0 & \text{otherwise,} \end{cases}$$

- $g_q(\mathbf{x}_i)$ is a "projection": i plays q if possible, else plays $\mathbf{x}_i$
- Implements a form of counteractual regret minimization
- In 3-player Kuhn poker, finds "best" equilibrium!

CFR-BR

Refs: [Johanson et al. ’11][Johanson et al. ’12]



- Minimize regret against a best responder
- Best responder uses *full unabstracted space*
- Use accelerated algorithms for computing best response
- Used in diabetes patient simulation [Chen & Bowling ’12]

CFR+

Refs: [Tammelin et al. '11]

Regret matching plus (RM$^+$): never accumulate negative regret!

CFR+

Refs: [Tammelin et al. ’11]

Regret matching plus (RM$^+$): never accumulate negative regret!

Theorem 1: T steps: RM$^+$ has external regret $\Delta_u \sqrt{|A|T}$.

CFR+

Refs: [Tammelin et al. '11]

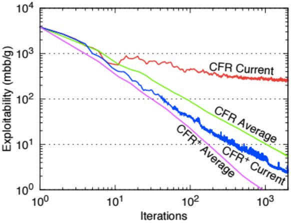
Regret matching plus (RM$^+$): never accumulate negative regret!

Theorem 1: T steps: RM$^+$ has external regret $\Delta_u \sqrt{|A|T}$.

Tracking regret [Herbster & Warmuth '98]: hindsight strategy can change $(k-1)$ times.

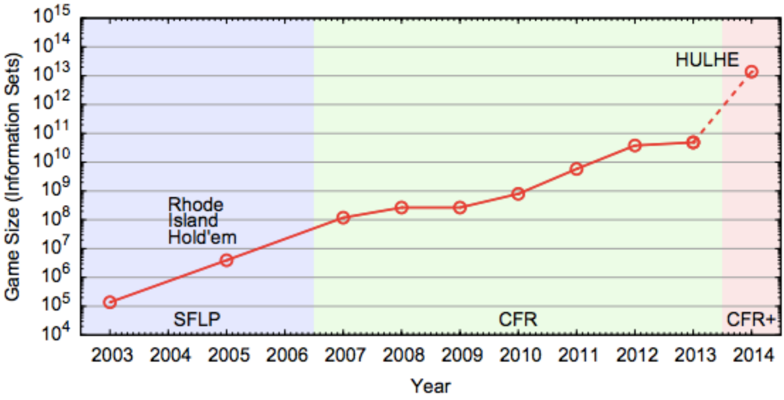
Theorem 2: T step: RM$^+$ has tracking regret $k\Delta_u \sqrt{|A|T}$.

Theorem 3: T step: CFR$^+$ has regret $O(|\mathcal{I}_1| + |\mathcal{I}_2|)\sqrt{|A|T}$.



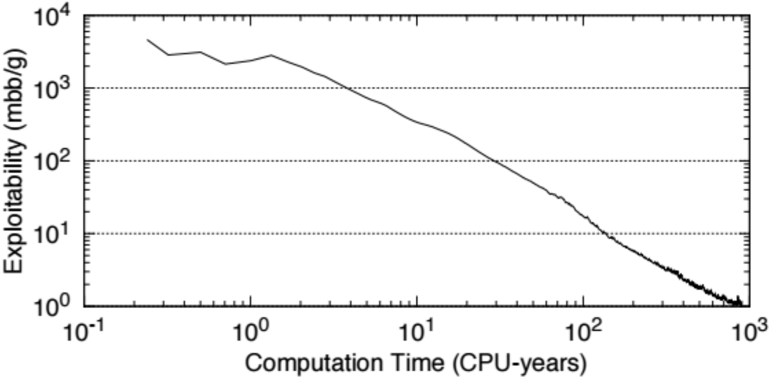
Solving 2-player HULHE

Refs: [Bowling et al. '15]



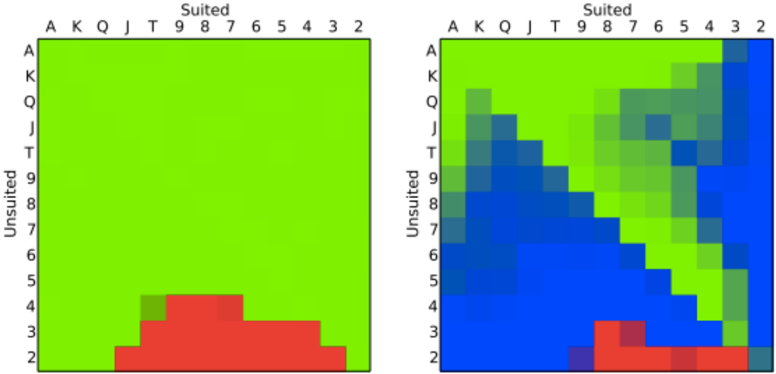
Solving 2-player HULHE

Refs: [Bowling et al. '15]



Solving 2-player HULHE

Refs: [Bowling et al. '15]



(a) first action as the dealer

(b) first action as the non-dealer after a dealer raise

Legend: fold raise call

Open Problem #1: Stronger-than-Nash?

Can a new variant of CFR converge to a:

- Sequential equilibrium?
- Trembling-hand perfect equilibrium?
- Strong equilibrium?

Open Problem #2: Correlated Equilibrium?

Does/can CFR converge to an (extensive-form) correlated equilibrium?

Other work

- FSICFR (chance-sampling variant) [Neller & Hnath '11]
- CFR with decomposition [Burch et al. '12][Jackson '14]
- Regret transfer [Brown and Sandholm '14]
- Regret-based Pruning [Brown and Sandholm '14]
- Automated abstraction and solving [Brown and Sandholm '15]
- Warm starting CFR [Brown and Sandholm '16]
- Online search [Lisý, Lanctot, and Bowling '15][Heinrich & Silver '15]
- Relationship to optimization [Waugh and Bagnell '15]
- Fictitious Self-play [Heinrich, Lanctot, and Silver '15]
- End-to-end learning [Waugh et al. '15][Heinrich and Silver '16]
- Application to security domains [Lisy, Davis, and Bowling '16]
- ...

Thanks, Questions, Info

Thank you for listening! Any questions?

Part 1: Sam Ganzfried

`sam.ganzfried@gmail.com`

`http://www.ganzfriedresearch.com`

Part 2: Marc Lanctot

`marc.lanctot@gmail.com`

`http://mlanctot.info`

`http://mlanctot.info/ecpokertutorial2016`