Strong Game-Theoretic Strategies: Beyond Two Agents

Sam Ganzfried

Assistant Professor, Computer Science, Florida International University, Miami FL

PhD, Computer Science Department, Carnegie Mellon University, 2015

sam.ganzfried@gmail.com

www.ganzfriedresearch.com

New title: How poker can help for hurricane evacuation

New title: How poker can help for hurricane evacuation

NOTE: You will need to book with your voucher before the expiration date, but you **DO NOT** need to travel by then - your travel dates can be later. Additional terms and conditions apply.*

For more info on how to use your voucher, **click here**.

Record Locator:KHLSGE

Voucher Number	Issued To	Voucher Amount	Expiration Date
24462044436300203	SAMUEL GANZFRIED	$50	11/04/2017

Jacksonville "Scramble"

bestbet
BOUNTY SCRAMBLE

WPT®
WORLD POKER TOUR

Milestones

- Opponent Modelling in Poker, Billings et.al., ‘98
- Abstraction Methods for Game-Theoretic Poker, Shi/Littman, ‘00
- Approximating Game-Theoretic Optimal Strategies for Full-scale Poker, Billings et.al., ‘03
- Optimal Rhode Island Poker, Gilpin/Sandholm, ‘05
- Annual Computer Poker Competition ‘06-Present
- EGT/automated abstraction algorithms, Gilpin/Sandholm ‘06-‘08
- Regret Minimization in Games with Incomplete Information, Zinkevich et.al., ‘07
- Man vs. Machine limit Texas hold ‘em competitions ‘08-’09
- Computer Poker & Imperfect Information Symposium/Workshop ‘12-Present
- Heads-up Limit Hold'em Poker is Solved, Bowling et.al., ‘15
- Brains vs. AI no-limit Texas hold ‘em competition ’15
- First Computer Poker Tutorial ‘16
- DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker ’17
- Second Brains vs. AI no-limit Texas hold ‘em competition ‘17

The main mathematical result is the proof of the existence in any game of at least one equilibrium point. Other results concern the geometrical structure of the set of equilibrium points of a game with a solution, the geometry of sub-solutions, and the existence of a symmetrical equilibrium point in a symmetrical game.

As an illustration of the possibilities for application a treatment of a simple three-man poker model is included.

Table of Contents

Applications

The study of n-person games for which the accepted ethics of fair play imply non-cooperative playing is, of course, an obvious direction in which to apply this theory. And poker is the most obvious target. The analysis of a more realistic poker game than our very simple model should be quite an interesting affair.

Computing an Approximate Jam/Fold Equilibrium for 3-player No-Limit Texas Hold'em Tournaments *

Sam Ganzfried
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
sganzfri@cs.cmu.edu

Tuomas Sandholm
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
sandholm@cs.cmu.edu

Poster Session: Tuesday 1:45 PM - 3:00 PM room **102**

How Can Form a Common Understanding of Price in G-Process
Presented by: YI Gan, Southwest University

Intertemporal Tradeoffs in Coordination Problems
Presented by: Jakub Steiner, The University of Edinburgh

Market research and complementary advertising under asymmetric information
Presented by: Toshihiro Tsuchihashi, Hitotsubashi university (grad student)

A New Concept of Solution for Fuzzy Matrix Games
Presented by: Moussa Larbani, IIUM University

Competition with Asymmetric Switching Costs
Presented by: Sebastian Infante Bilbao, Universidad de Chile

Algorithms for Multiplayer Stochastic Games of Imperfect Information with Application to Three-Player No-Limit Texas Hold'em Tournaments
Presented by: Sam Ganzfried, Carnegie Mellon University

An evolutionary argument for inequity aversion
Presented by: Robertas Zubrickas, Stockholm School of Economics

Expert Advice and Amateur Interpretations
Presented by: Ernest Lai, University of Pittsburgh

Scientific Collaboration Networks: The role of Heterogeneity and Congestion
Presented by: Antoni Rubí-Barceló, Universitat Pompeu Fabra

Games in the Eurasian gas supply network:
Presented by: Svetlana Ikonnikova, Catholic University of Leuven

Cardinal Bayesian Nontransfer Allocation Mechanisms. The Two-Object Case
Presented by: Antonio Miralles, Boston University

Natural Oligopoly in Industrial Research Collaboration
Presented by: Bastian Westbrock, Utrecht University

Scope and applicability of game theory

- Strategic multiagent interactions occur in all fields
  - Economics and business: bidding in auctions, offers in negotiations
  - Political science/law: fair division of resources, e.g., divorce settlements
  - Biology/medicine: robust diabetes management (robustness against "adversarial" selection of parameters in MDP)
  - Computer science: theory, AI, PL, systems; national security (e.g., deploying officers to protect ports), cybersecurity (e.g., determining optimal thresholds against phishing attacks), internet phenomena (e.g., ad auctions)

Game theory background

	rock	paper	scissors
Rock	0,0	-1, 1	1, -1
Paper	1,-1	0, 0	-1,1
Scissors	-1,1	1,-1	0,0

- Players
- Actions (aka pure strategies)
- Strategy profile: e.g., (R,p)
- Utility function: e.g., $u_1(R,p) = -1$, $u_2(R,p) = 1$

Zero-sum game

	rock	paper	scissors
Rock	0,0	-1, 1	1, -1
Paper	1,-1	0, 0	-1,1
Scissors	-1,1	1,-1	0,0

- Sum of payoffs is zero at each strategy profile: e.g., $u_1(R,p) + u_2(R,p) = 0$
- Models purely adversarial settings

Mixed strategies

- Probability distributions over pure strategies
- E.g., R with prob. 0.6, P with prob. 0.3, S with prob. 0.1

Best response (aka nemesis)

- Any strategy that maximizes payoff against opponent's strategy
- If P2 plays (0.6, 0.3, 0.1) for r,p,s, then a best response for P1 is to play P with probability 1

Nash equilibrium

- Strategy profile where all players simultaneously play a best response
- Standard solution concept in game theory
  - Guaranteed to always exist in finite games [Nash 1950]
- In Rock-Paper-Scissors, the unique equilibrium is for both players to select each pure strategy with probability 1/3

Minimax Theorem

- Minimax theorem: For every two-player zero-sum game, there exists a value v* and a mixed strategy profile σ^* such that:
  a. P1 guarantees a payoff of at least v* in the worst case by playing σ^*_1
  b. P2 guarantees a payoff of at least -v* in the worst case by playing σ^*_2
- v* (= v_1) is the *value* of the game
- All equilibrium strategies for player i guarantee at least v_i in the worst case
- For RPS, v* = 0

Exploitability

- Exploitability of a strategy is difference between value of the game and performance against a best response
  - Every equilibrium has zero exploitability
- Always playing rock has exploitability 1
  - Best response is to play paper with probability 1

Nash equilibria in two-player zero-sum games

- Zero exploitability – “unbeatable”
- Exchangeable
  - If (a,b) and (c,d) are NE, then (a,d) and (c,b) are too
- Can be computed in polynomial time by a linear programming (LP) formulation

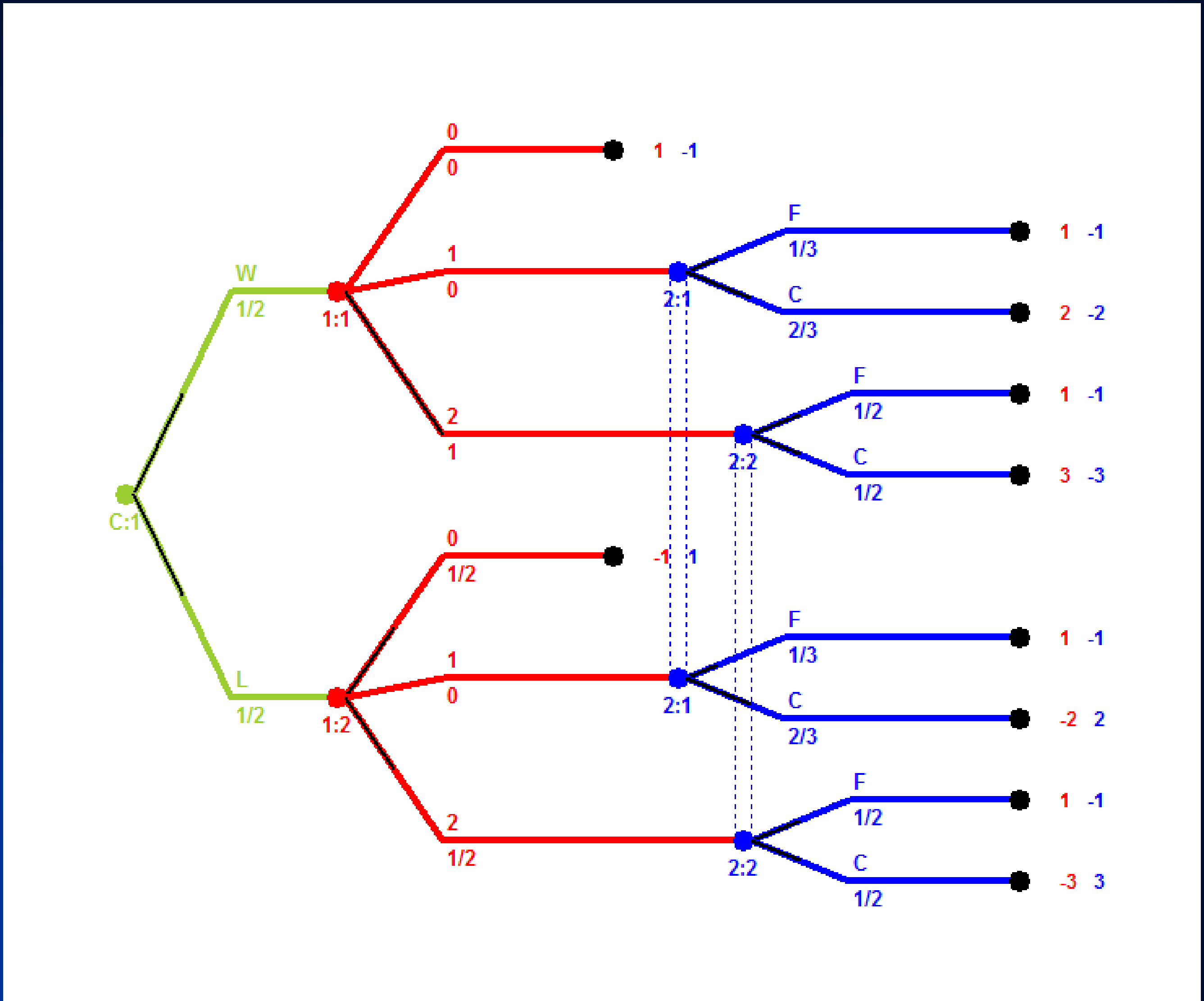
Nash equilibria in multiplayer and non-zero-sum games

- None of the two-player zero-sum results hold
- There can exist multiple equilibria, each with different payoffs to the players
- If one player follows one equilibrium while other players follow a different equilibrium, overall profile is not guaranteed to be an equilibrium
- If one player plays an equilibrium, he could do worse if the opponents deviate from that equilibrium
- Computing an equilibrium is PPAD-hard

Imperfect information

- In many important games, there is information that is private to only some agents and not available to other agents
  - In auctions, each bidder may know his own valuation and only know the distribution from which other agents' valuations are drawn
  - In poker, players may not know private cards held by other players

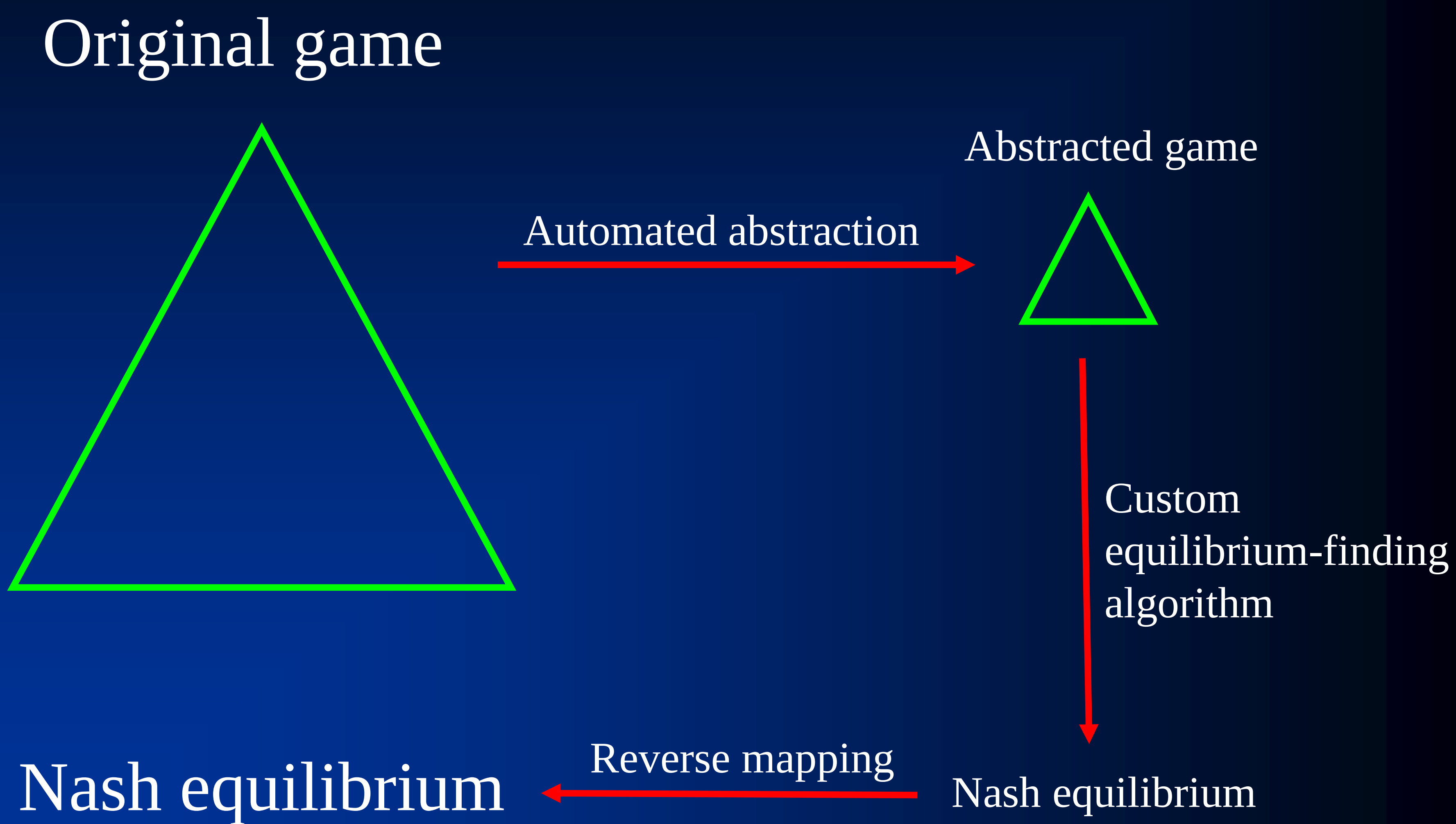
Extensive-form representation



Extensive-form games

- Two-player zero-sum EFGs can be solved in polynomial time by linear programming
  - Scales to games with up to 10^8 states
- Iterative algorithms (CFR and EGT) have been developed for computing an ε-equilibrium that scale to games with 10^{17} states
  - CFR also applies to multiplayer and general sum games, though no significant guarantees in those classes
  - (MC)CFR is self-play algorithm that samples actions down tree and updates regrets and average strategies stored at every information set

Standard paradigm for solving large imperfect-information games



Texas hold ‘em poker

- Huge game of imperfect information
  - Most studied imp-info game in AI community since 2006 due to AAAI computer poker competition
  - Most attention on 2-player variants (2-player zero-sum)
  - Multi-billion dollar industry (not “frivolous”)
- Limit Texas hold ‘em – fixed betting size
  - $\sim 10^{17}$ nodes in game tree
- No Limit Texas hold ‘em – unlimited bet size
  - $\sim 10^{165}$ nodes in game tree
  - Most active domain in last several years
  - Most popular variant for humans

No-limit Texas hold ‘em poker

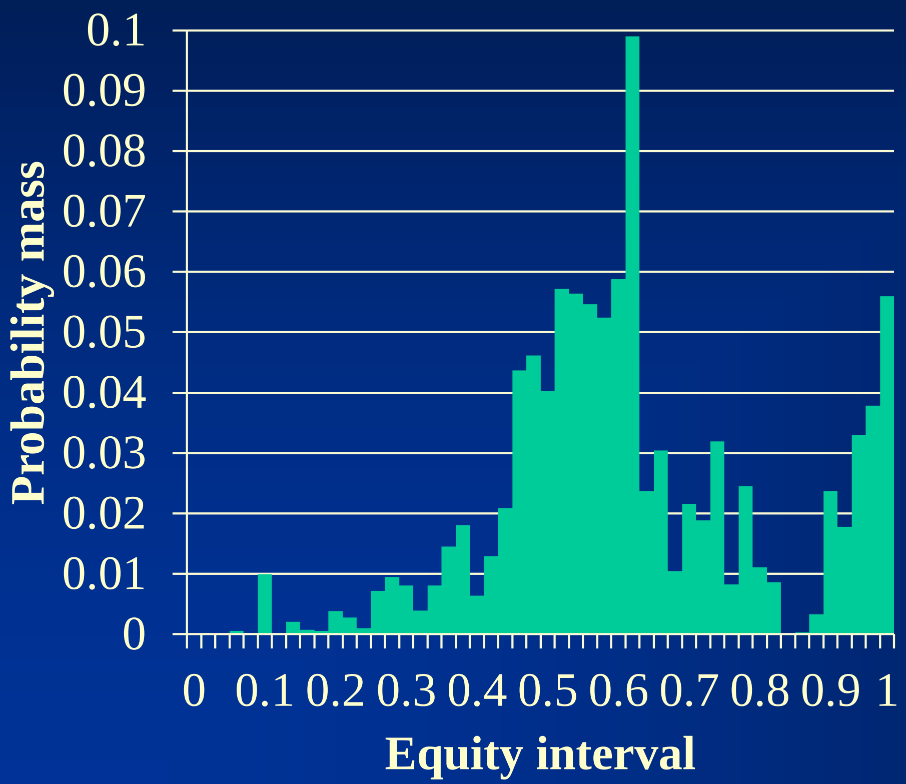
- Two players have stack and pay blinds (ante)
- Each player dealt two private cards
- Round of betting (preflop)
  - Players can fold, call, bet (any amount up to stack)
- Three public cards dealt (flop) and a second round of betting
- One more public card and round of betting (turn)
- Final card and round of betting (river)
- Showdown

Game abstraction

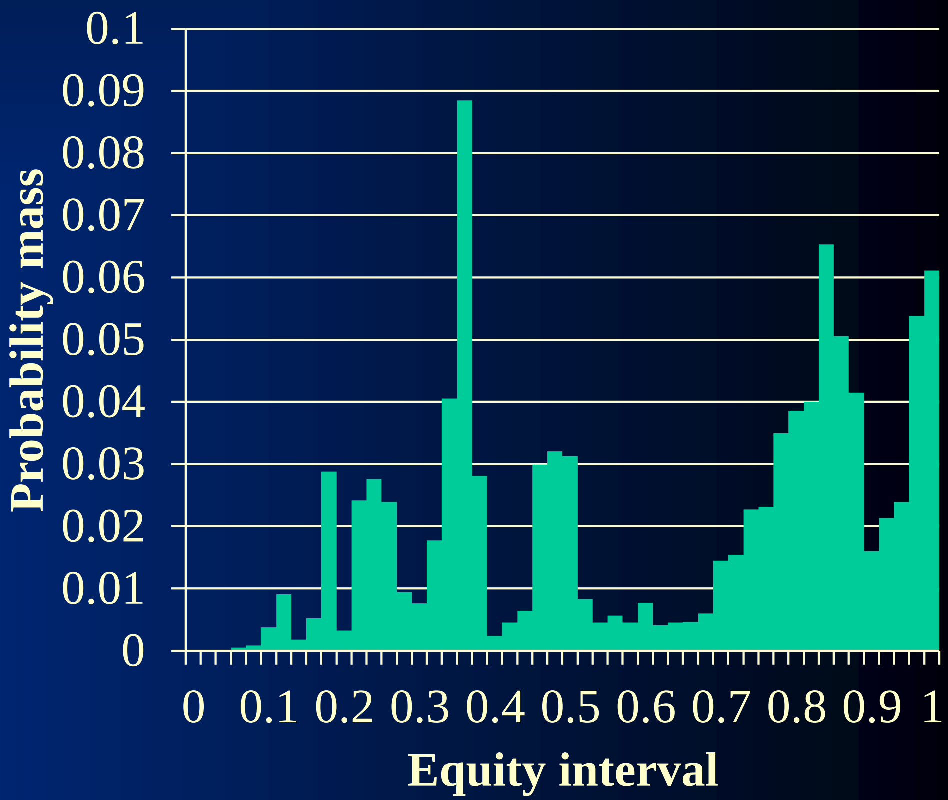
- Necessary for solving large games
  - 2-player no-limit Texas hold 'em has 10^{165} game states, while best solvers "only" scale to games with 10^{17} states
- Information abstraction: grouping information sets together
- Action abstraction: discretizing action space
  - E.g., limit bids to be multiples of \$10 or \$100

Information abstraction

Equity distribution for 6c6d. EHS: 0.634



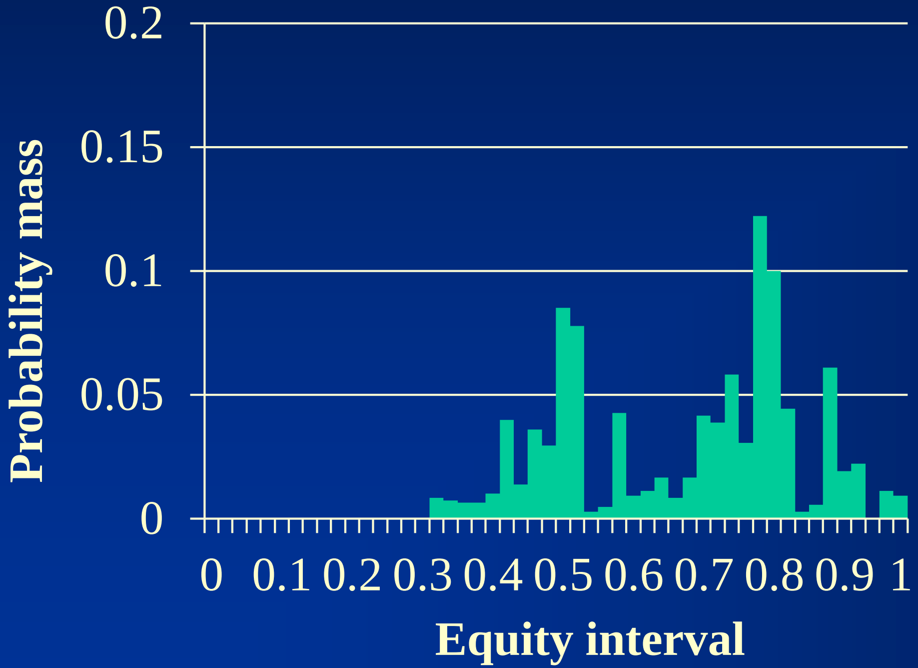
Equity distribution for KcQc. EHS: 0.633



Potential-aware abstraction with EMD

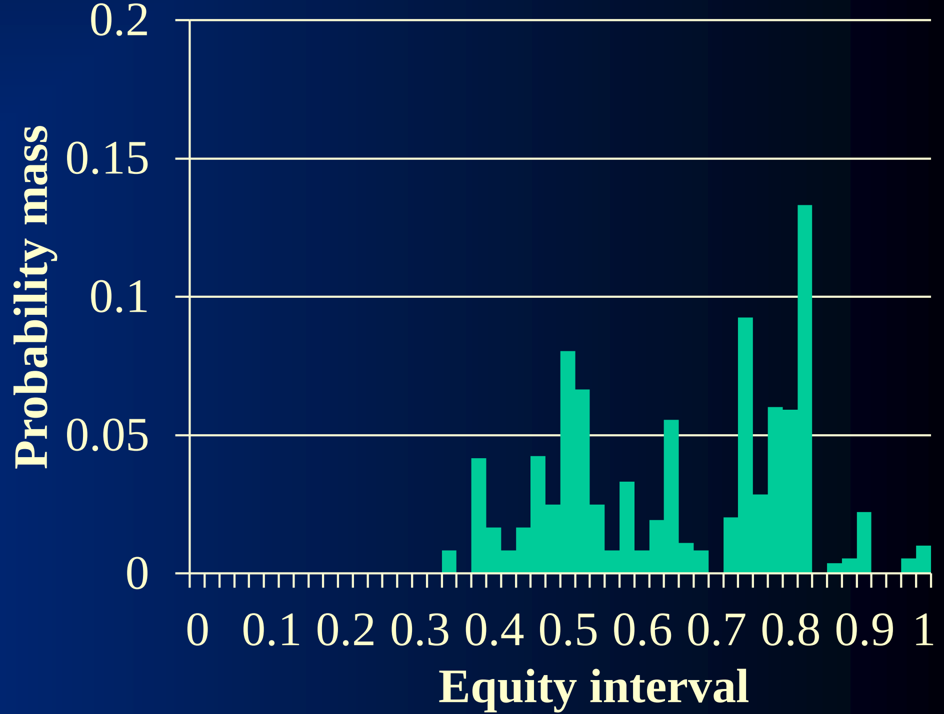
Equity distribution for TcQd-7h9hQh on river (final round)

EHS: 0.683



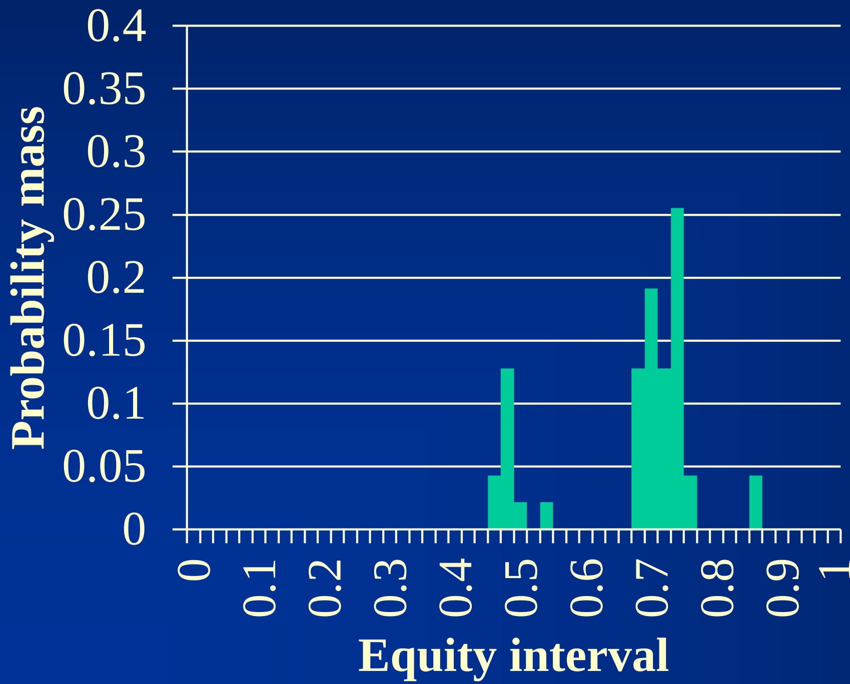
Equity distribution for 5c9d-3d5d7d on river (final round)

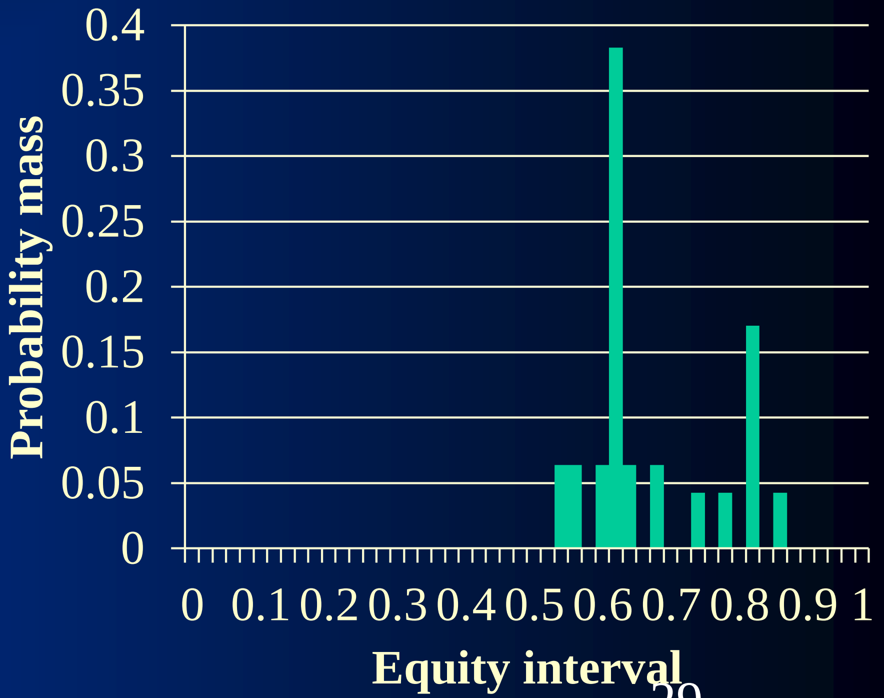
EHS: 0.679



Potential-aware abstraction with EMD

- Equity distributions on the turn. Each point is EHS for given turn card assuming uniform random river and opponent hand
- EMD is 4.519 (vs. 0.559 using comparable units to river EMD)

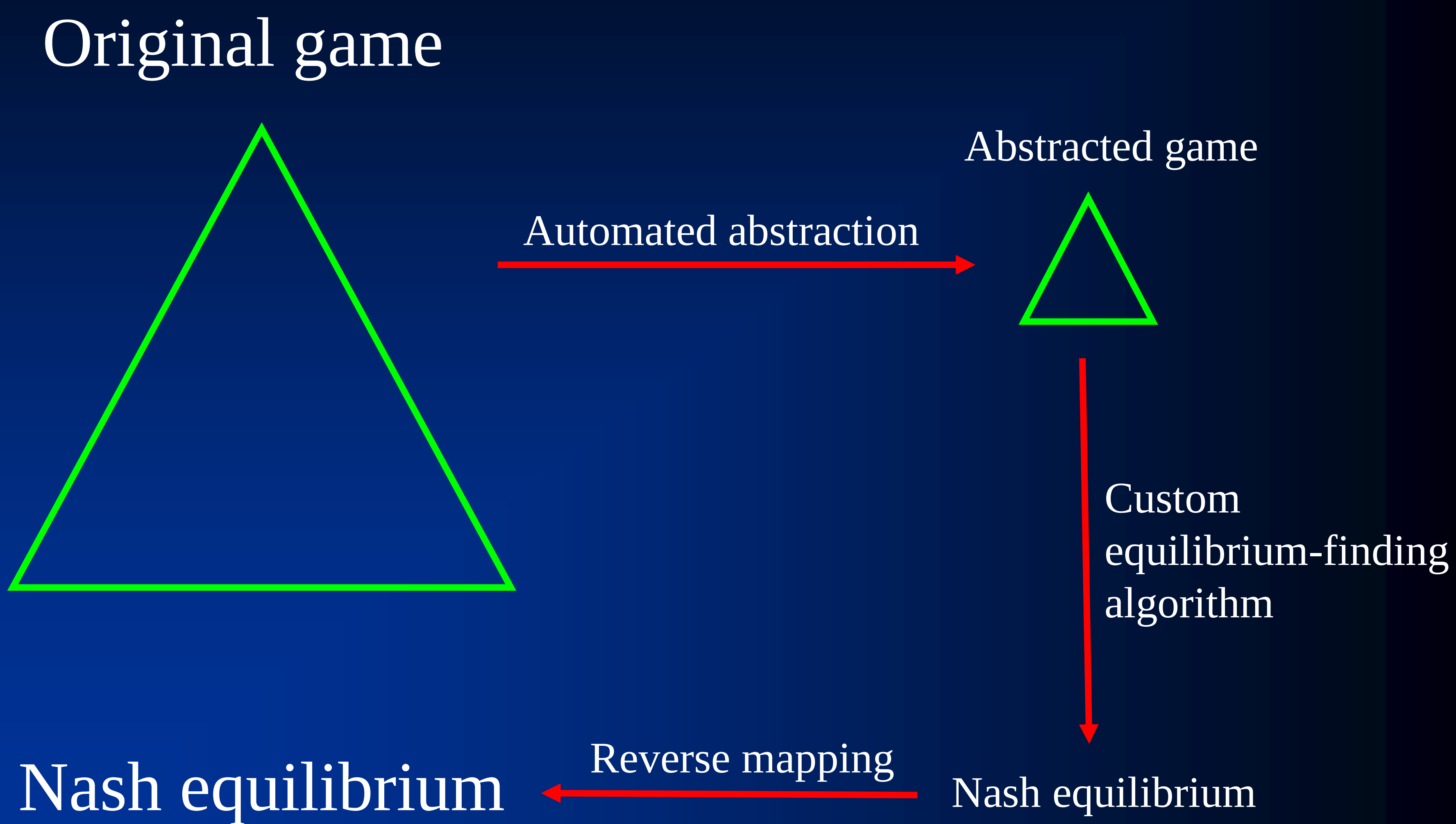




Algorithm for potential-aware imperfect-recall abstraction with EMD

- Bottom-up pass of the information tree (assume an abstraction for final rounds has already been computed using arbitrary approach)
- For each round n
  - Let m^{n+1}_i denote mean of cluster i in A^{n+1}
  - For each pair of round n+1 clusters (i,j), compute distance $d^n_{i,j}$ between m^{n+1}_i and m^{n+1}_j using d^{n+1}
  - For each point x^n, create histogram over clusters from A^{n+1}
  - Compute abstraction A^n using EMD with $d^n_{i,j}$ as ground distance function
    - Developed fast custom heuristic for approximating EMD in our multidimensional setting
    - Best commercially-available algorithm was far too slow to compute abstractions in poker

Standard paradigm for solving large extensive-form games



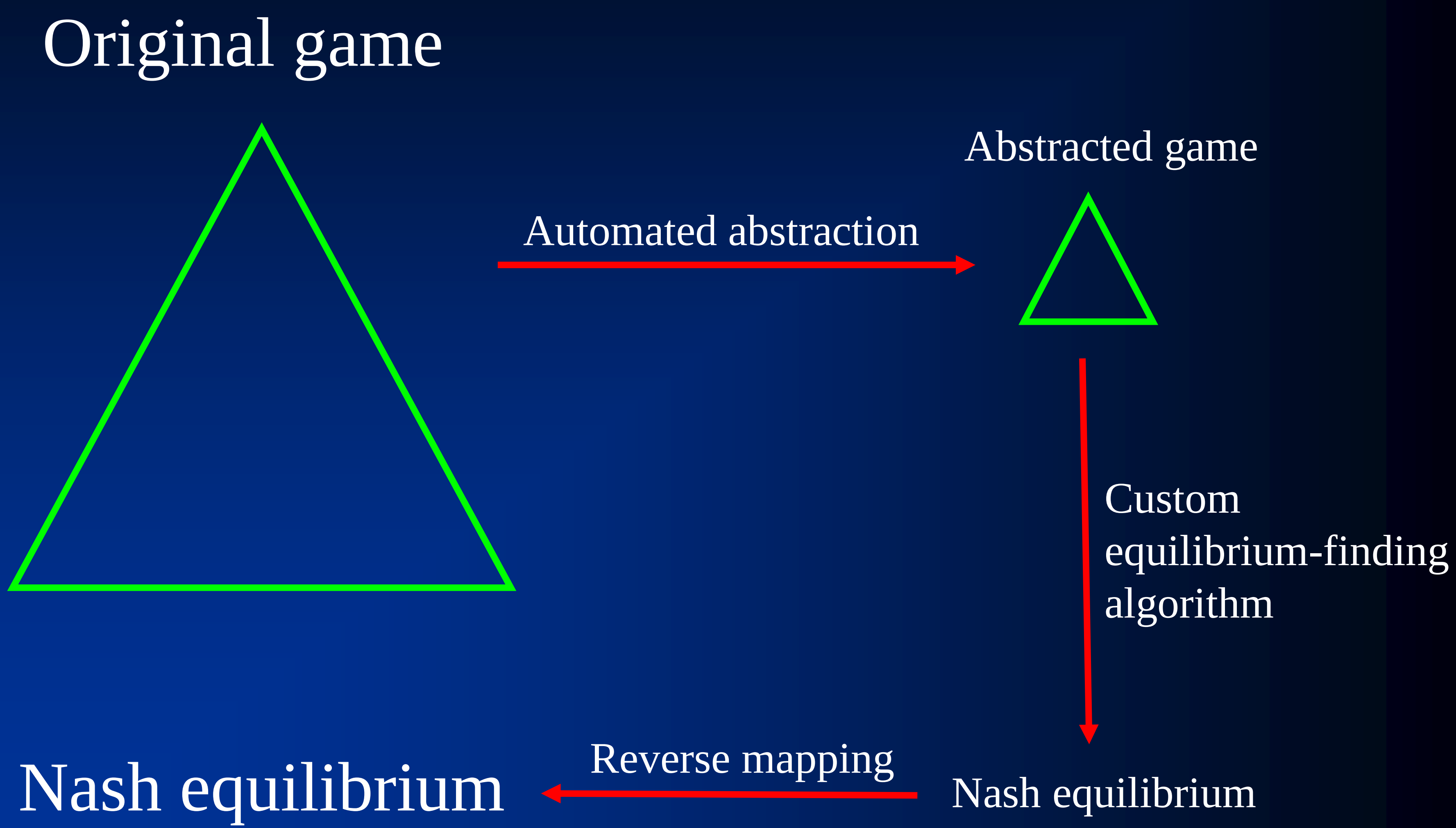
Hierarchical abstraction to enable distributed equilibrium computation

- On distributed architectures and supercomputers with high inter-blade memory access latency, straightforward MCCFR parallelization approaches lead to impractically slow runtimes
  - When a core does an update at an information set it needs to read and write memory with high latency
  - Different cores working on same information set may need to lock memory, wait for each other, possibly over-write each others' parallel work, and work on out-of-sync inputs
- Our approach solves the former problem and also helps mitigate the latter issue

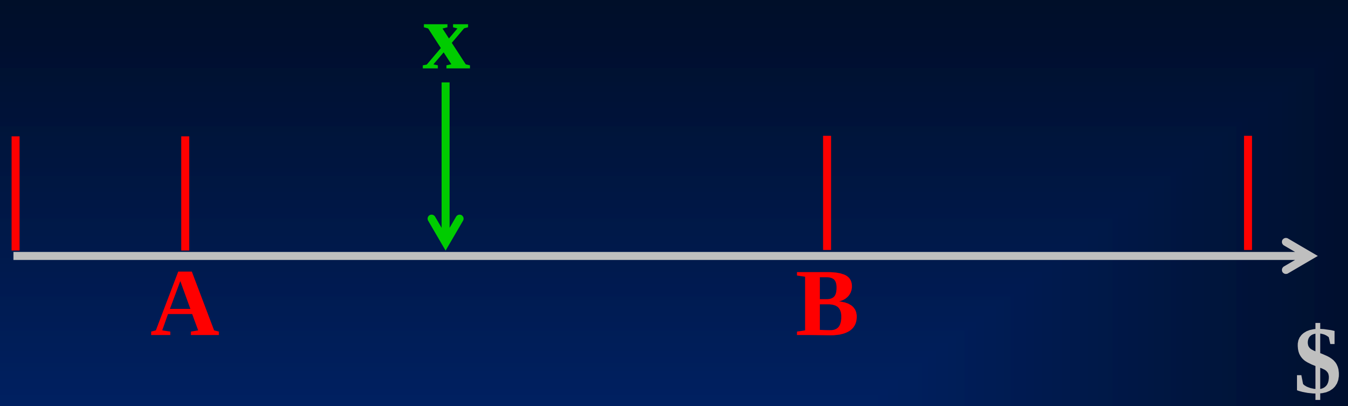
High-level approach

- To obtain these benefits, our algorithm creates an information abstraction that allows us to assign disjoint components of the game tree to different blades so the trajectory of each sample only accesses information sets located on the same blade.
  - First cluster public information at some early point in the game (public flop cards in poker), then cluster private information separately for each public cluster.
- Run modified version of external-sampling MCCFR
  - Samples one pair of preflop hands per iteration. For the later betting rounds, each blade samples public cards from its public cluster and performs MCCFR within each cluster.

Standard paradigm for solving large imperfect-information games



Action translation



- $f_{A,B}(x) \equiv$ probability we map x to A
  - Will also denote as just $f(x)$

[Ganzfried & Sandholm IJCAI-13]

A natural approach

- If x < $\frac{A+B}{2}$, then map x to A; otherwise, map x to B
- Called the **deterministic arithmetic** mapping

- Suppose pot is 1, stacks are 100
- Suppose we are using the {fold, call, pot, all-in} action abstraction
  - “previous expert knowledge [has] dictated that if only a single bet size [in addition to all-in] is used everywhere, it should be pot sized” [Hawkin et al., AAAI 2012]
- Suppose opponent bets x in (1,100)
  - So A = 1, B = 100

- Suppose we call a bet of 1 with probability ½ with a medium-strength hand
- Suppose the opponent has a very strong hand
- His expected payoff of betting 1 will be:

  $(1 \cdot \frac{1}{2}) + (2 \cdot \frac{1}{2}) = 1.5$
- If instead he bets 50, his expected payoff will be:

  $(1 \cdot \frac{1}{2}) + (51 \cdot \frac{1}{2}) = 26$
- He gains $24.50 by exploiting our translation mapping!
- Tartanian1 lost to an agent that didn't look at its private cards in 2007 ACPC using this mapping!

An improvement

- What if we randomize and map x to A with probability $\frac{B-x}{B-A}$?
- Suppose opponent bets 50.5, and we call an all-in bet with probability $\frac{1}{101}$ with a mediocre hand
- Then his expected payoff is \$13.875
- An improvement, but still way too high
- Called the **randomized arithmetic** mapping

Other prior approaches

- Deterministic geometric: If $\frac{A}{x} > \frac{x}{B}$, map x to A; otherwise, map x to B
  - Used by Tartanian2 in 2008
- Randomized geometric 1
  - $f(x) = \frac{A(B-x)}{A(B-x) + x(x-A)}$
  - Used by Alberta 2009-present
- Randomized geometric 2
  - $f(x) = \frac{A(B+x)(B-x)}{(B-A)(x^2 + AB)}$
  - Used by CMU 2010-2011

Problem with prior approaches?

- High exploitability in simplified games
- Purely heuristic and not based on any theoretical justification
- Fail to satisfy natural theoretical properties

Our new mapping

- We propose a new mapping, called the pseudo-harmonic mapping, which is the only mapping consistent with the equilibrium of a simplified poker game:
  - $f(x) = \frac{(B-x)(1+A)}{(B-A)(1+x)}$
- This mapping has significantly lower exploitability than the prior ones in several simplified poker games
- Significantly outperforms the randomized-geometric mappings in no-limit Texas hold'em

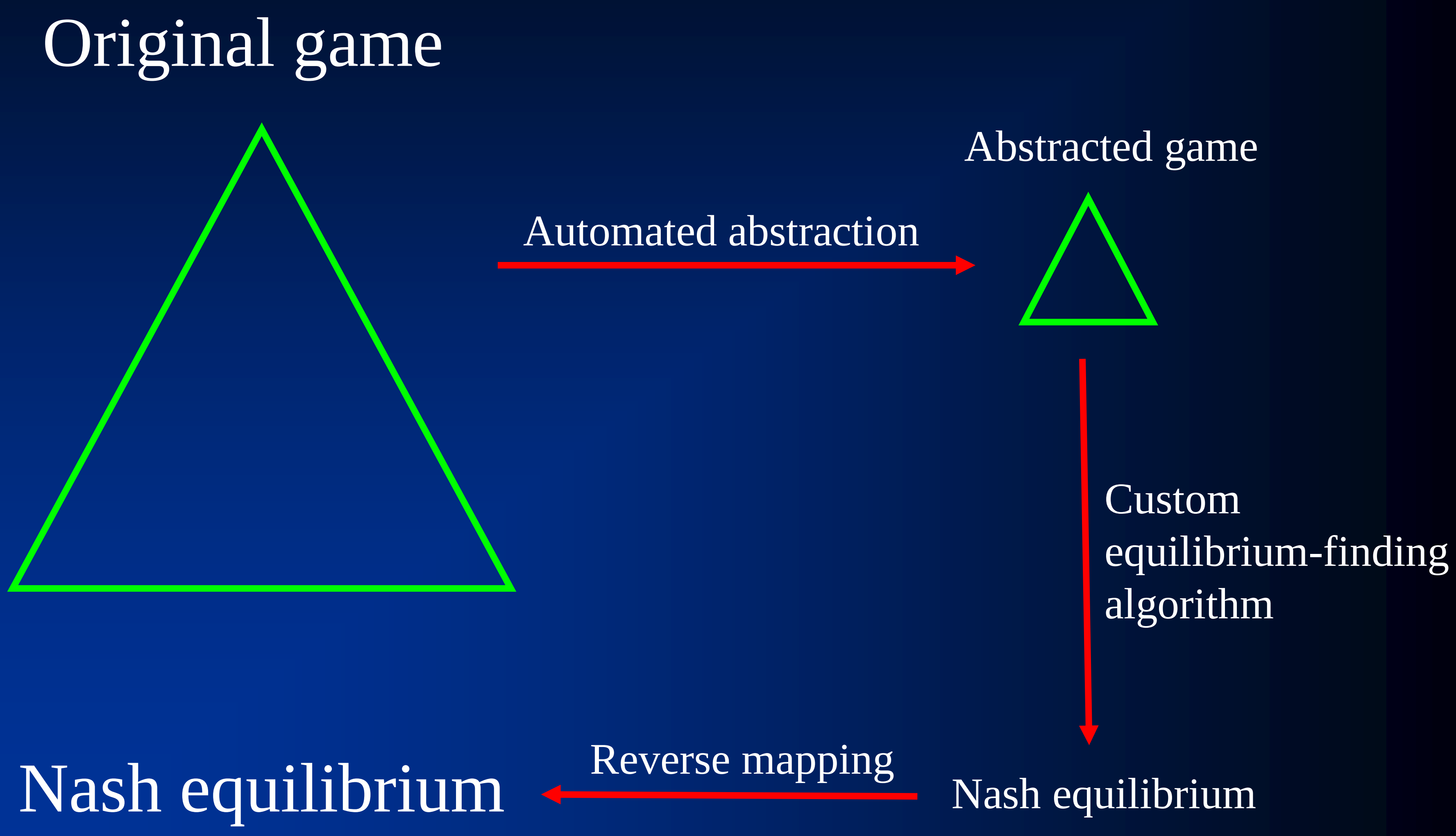
Action translation desiderata

1. Boundary constraints: f(A) = 1, f(B) = 0
2. Monotonicity
3. Scale invariance
4. Action robustness: small change in x doesn’t lead to large change in f
5. Boundary robustness: small change in A or B doesn’t lead to large change in f

Theoretical results

- Randomized geometric mappings violate boundary robustness. If we allow A = 0 they are discontinuous in A. Otherwise, they are Lipschitz-discontinuous in A.
- Only randomized-arithmetic and randomized-pseudo-harmonic satisfy all the desiderata

Standard paradigm for solving large imperfect-information games



Purification and thresholding

- *Thresholding*: round action probabilities below c down to 0 (then renormalize)
- *Purification* is extreme case where we play maximal-probability action with probability 1

Experiments on no-limit Texas hold ‘em

- Purification outperforms using a threshold of 0.15
  - Does better than it against all but one 2010 competitor, beats it head-to-head, and won bankroll competition

Worst-case exploitability

- We also compared *worst-case exploitabilities* of several variants submitted to the 2010 two-player limit Texas hold 'em division
  - Using algorithm of Johanson *et al.* IJCAI-11

Threshold	Exploitability of GS6	Exploitability of Hyperborean
None	463.591	**235.209**
0.05	326.119	243.705
0.15	**318.465**	258.53
0.25	335.048	277.841
Purified	349.873	437.242

Table 4: Results for full-game worst-case exploitabilities of several strategies in two-player limit Texas Hold'em. Results are in milli big blinds per hand. Bolded values indicate the lowest exploitability achieved for each strategy.

Purification and thresholding

- 4x4 two-player zero-sum matrix games with payoffs uniformly at random from [-1,1]
- Compute equilibrium F in full game
- Compute equilibrium A in abstracted game that omits last row and column
  - essentially "random" abstractions
- Compare $u_1(A_1, F_2)$ to $u_1(pur(A_1), F_2)$
- **Conclusion: Abstraction+purification outperforms just abstraction (against full equilibrium) at 95% confidence level**

Purification and thresholding

Purified average payoff	-0.050987 +- 0.00042
Unpurified average payoff	-0.054905 +- 0.00044
# games where purification led to improved performance	261569 (17.44 %)
# games where purification led to worse performance	172164 (11.48%)
# games where purification led to no change in performance	1066267 (71.08 %)

- Some conditions when they perform identically:
  1. The abstract equilibrium A is a pure strategy profile
  2. The support of A_1 is a subset of the support of F_1

Purification and thresholding

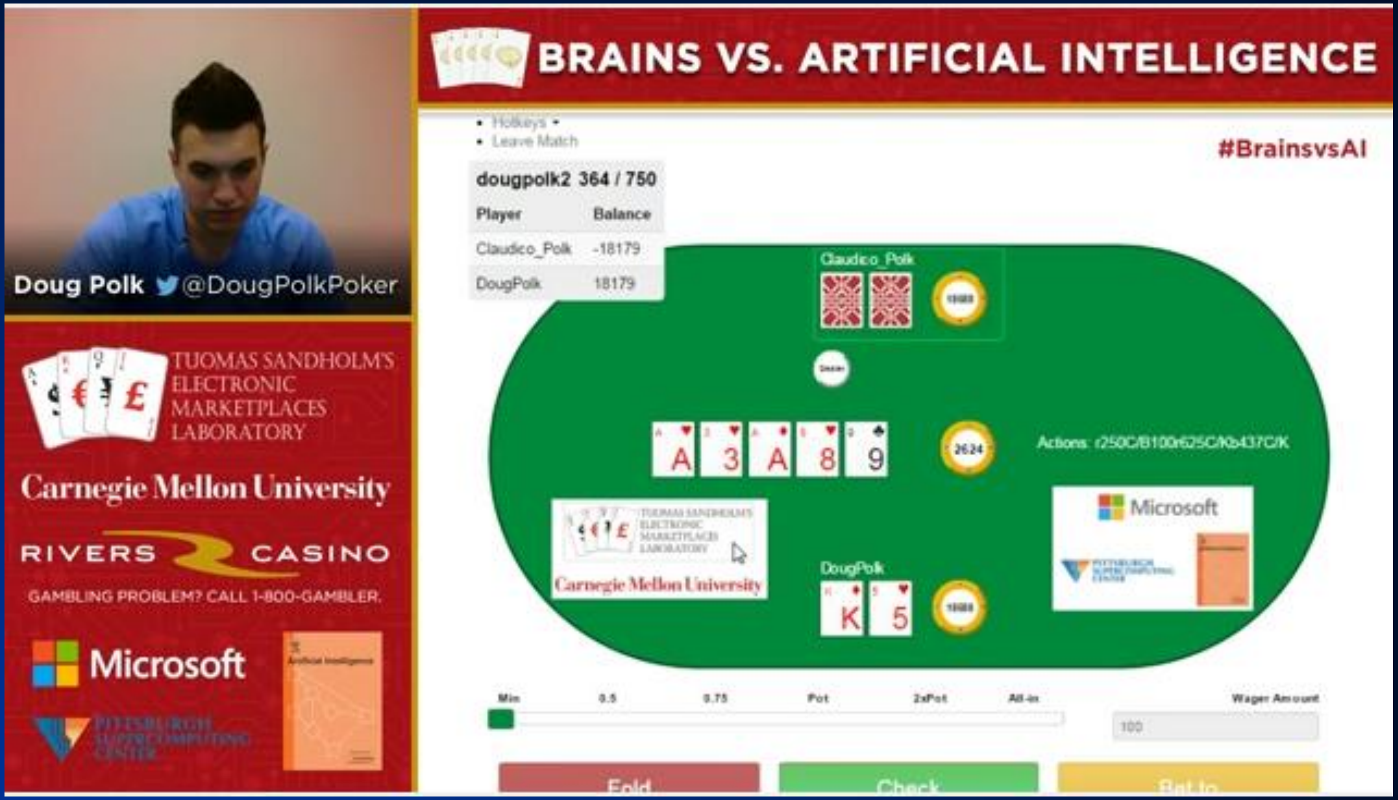
- Results depend crucially on the support of the full equilibrium
- If we only consider the set of games that have an equilibrium σ with a given support, purification improves performance for each class except for the following, where the performance is statistically indistinguishable:
  - σ is the pure strategy profile in which each player plays his fourth pure strategy
  - σ is a mixed strategy profile in which player 1's support contains his fourth pure strategy, and player 2's support does not contain his fourth pure strategy

New family of post-processing techniques

- 2 main ideas:
  - Bundle similar actions
  - Add preference for conservative actions
- First separate actions into {fold, call, “bet”}
  - If probability of folding exceeds a threshold parameter, fold with prob. 1
  - Else, follow purification between fold, call, and “meta-action” of “bet.”
  - If “bet” is selected, then follow purification within the specific bet actions.
- Many variations: threshold parameter, bucketing of actions, thresholding value among buckets, etc.

Post-processing experiments

	Hyperborean.iro	Slumbot	Average	Min
No Thresholding	+30 ± 32	+10 ± 27	+20	+10
Purification	+55 ± 27	+19 ± 22	+37	+19
Thresholding-0.15	+35 ± 30	+19 ± 25	+27	+19
New-0.2	+39 ± 26	+103 ± 21	+71	+39


BRAINS VS. ARTIFICIAL INTELLIGENCE
Doug Polk @DougPolkPoker
TUOMAS SANDHOLM'S ELECTRONIC MARKETPLACES LABORATORY
Carnegie Mellon University
RIVERS CASINO
GAMBLING PROBLEM? CALL 1-800-GAMBLER.
Microsoft
#BrainsvsAI
dougpolk2 364 / 750
Player Balance
Claudico_Polk -18179
DougPolk 18179
Claudico_Polk
DougPolk
Actions: r250C/B100r625C/Kb437C/K
Fold
Check
Bet to

Brains vs. Artificial Intelligence

- April 24-May 8, 2015 at Rivers Casino in Pittsburgh, PA
  - The competition was organized by Carnegie Mellon University Professor Tuomas Sandholm. Collaborators were Tuomas Sandholm and Noam Brown.
- 20,000 hands of two-player no-limit Texas hold ‘em between “Claudico” and Dong Kim, Jason Les, Bjorn Li, Doug Polk
  - 80,000 hands in total
- Used “duplicate” scoring

Brains

Doug Polk
@DougPolkPoker

March HUNL PR
1 West Coast Gangsters
2 Big Dick
3 AZNflushie (RIP)
4 Rumble man
5 Swarmmy
6 Kaby
7 Ike
8 wheyprotein
9 80%carry
10 muumi

RETWEETS 4 LIKES 13

4:24 AM - 28 Feb 2015

Nick Frame
@TCfromUB

Follow

The REAL power rankings for OCT 2014 are out

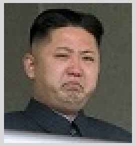
TC power rankings OCT 2014

1. WCG (0)
2. ike (+1)
3. sauce (+1)
4. TCfromUB (+1)
5. jungle (+5)
6. pandorasbux (-4)
7. kabydf (0)
8. donger (-2)
9. carrycakes (-1)
10. KPR (-1)
11. asianflushie (+3)
12. kanu7 (+3)
13. bajskorven (U)
14. OTBredbaron (U)
15. Rperfumo (-4)
16. mokoma1 (0)
17. Billiomucks (-5)
18. dougiedan (-5)
19. ForTheSwarm (U)
20. Willhasha (U)

Brains

02-04-2015, 09:53 AM #1

Donger Kim
enthusiast



Join Date: Feb 2015
Location: South Korea
@dongerkim
Posts: 54

Donger Kim to Nick Frame (TCfromUB) HU Challenge

I am a high-stakes heads up nlhe regular on PokerStars where I play under the name "Donger Kim". There's been quite a bit of discussion on heads-up rankings lately, particularly from TCfromUB (Nick Frame, TooCuriosso1 on 2p2). I've played quite a bit with him and think he's a top player. I respect his game and it would be humbling to play him and represent my country.

However, as he ranks himself ahead of me, I'd like to have a chance to play him in a challenge-type format. I think it would be a fun experience and something that would also be enjoyable for the community.

I propose we do a 15k hand challenge at 100/200 nl with a $50k sidebet escrowed with ike or sauce. I suggest we put some reasonable time frame conditions on this, we're both grinders so we should be able to finish this in a 1-2 week time frame.

Nick, let me know when you'd like to begin. Ideally, I'd like to get started right away.

Brains

MAR 2

Donger Kim wins heads-up challenge against TCfromUB

Poker News

G+1 0 Like 1 Share

Dong "Donger Kim" Kim won $103.992 from Nick "TCfromUB" Frame in the15,000 hand heads-up challenge, which not only earned him the respect of the high stakes community, but also an additional $15,000 from the sidebets for the challenge.

Results

- Humans won by 732,713 chips, which corresponds to 9.16 big blinds per 100 hands (BB/100) (SB = 50, BB = 100)
  – Statistically significant at 90% confidence level, but not 95% level
- Dong Kim beat Nick Frame by 13.87 BB/100
  – $103,992 over 15,000 hands with 25-50 blinds
- Doug Polk beat Ben Sulsky by 24.67 BB/100
  – $740,000 over 15,000 hands with 100-200 blinds

Payoffs

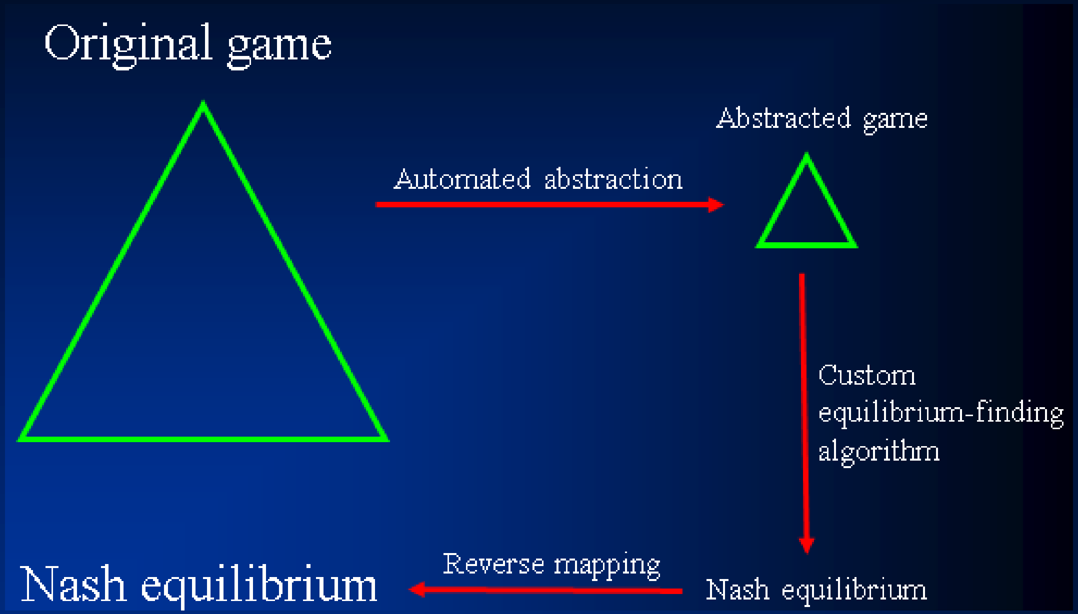
- Prize pool of \$100,000 distributed to the humans depending on their individual profits.

$$
\begin{aligned}
&\text{If } x_1 > x_4 \\
&\quad p_1 = \$10,000 + \$60,000 \cdot \frac{x_1 - x_4}{x_1 + x_2 + x_3 - 3x_4} \\
&\quad p_2 = \$10,000 + \$60,000 \cdot \frac{x_2 - x_4}{x_1 + x_2 + x_3 - 3x_4} \\
&\quad p_3 = \$10,000 + \$60,000 \cdot \frac{x_3 - x_4}{x_1 + x_2 + x_3 - 3x_4} \\
&\quad p_4 = \$10,000 \\
&\text{Else} \\
&\quad p_1 = p_2 = p_3 = p_4 = \$25,000
\end{aligned}
$$

I Limp!

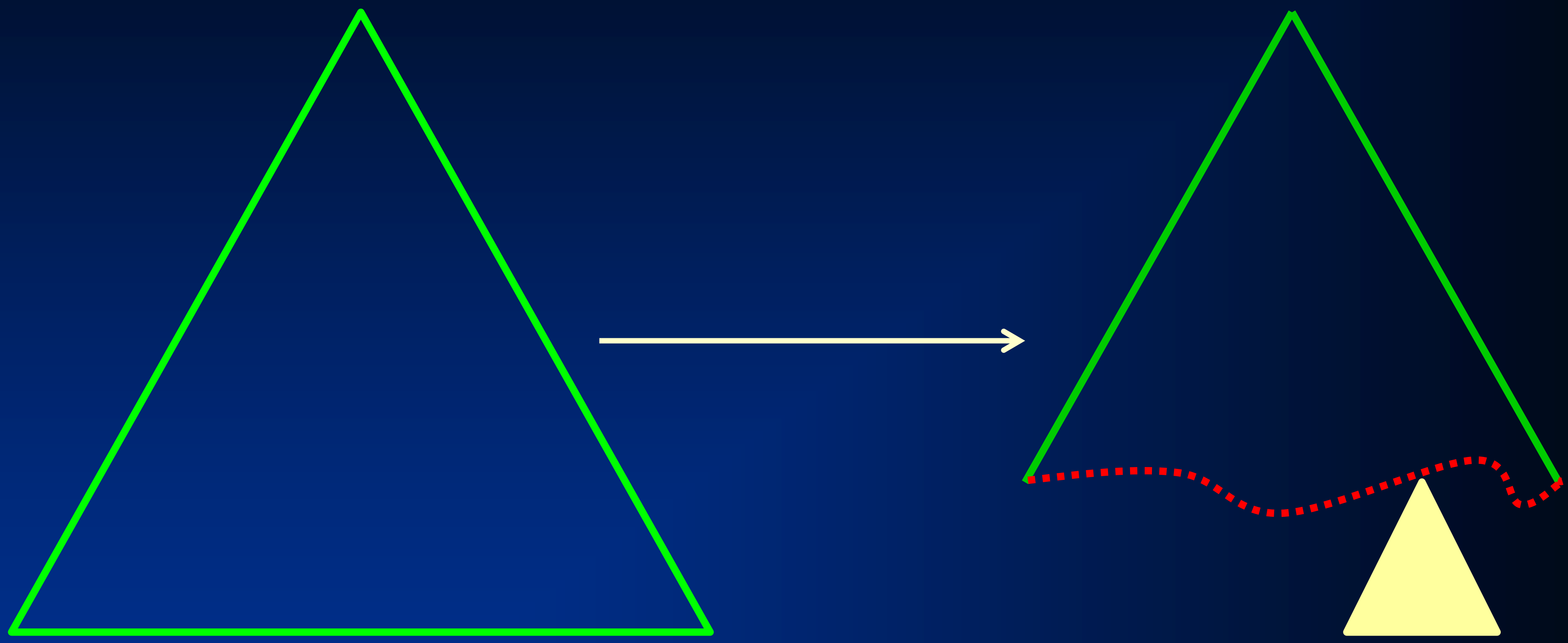
- "**Limping is for Losers**. This is *the most important fundamental* in poker -- for every game, for every tournament, every stake: If you are the first player to voluntarily commit chips to the pot, open for a raise. Limping is inevitably a losing play. If you see a person at the table limping, you can be fairly sure he is a bad player. Bottom line: If your hand is worth playing, it is worth raising" [Phil Gordon's Little Gold Book, 2011]
- Claudico limps close to 10% of its hands
  - Based on humans' analysis it profited overall from the limps
- Claudico makes many other unconventional plays (e.g., small bets of 10% pot and all-in bets for 40 times pot)

Architecture



- Offline abstraction and equilibrium computation
  - EC used Pittsburgh's Blacklight supercomputer with 961 cores
- Action translation
- Post-processing
- Endgame solving

Endgame solving



Strategies for entire game computed offline

Endgame strategies computed in real time to greater degree of accuracy

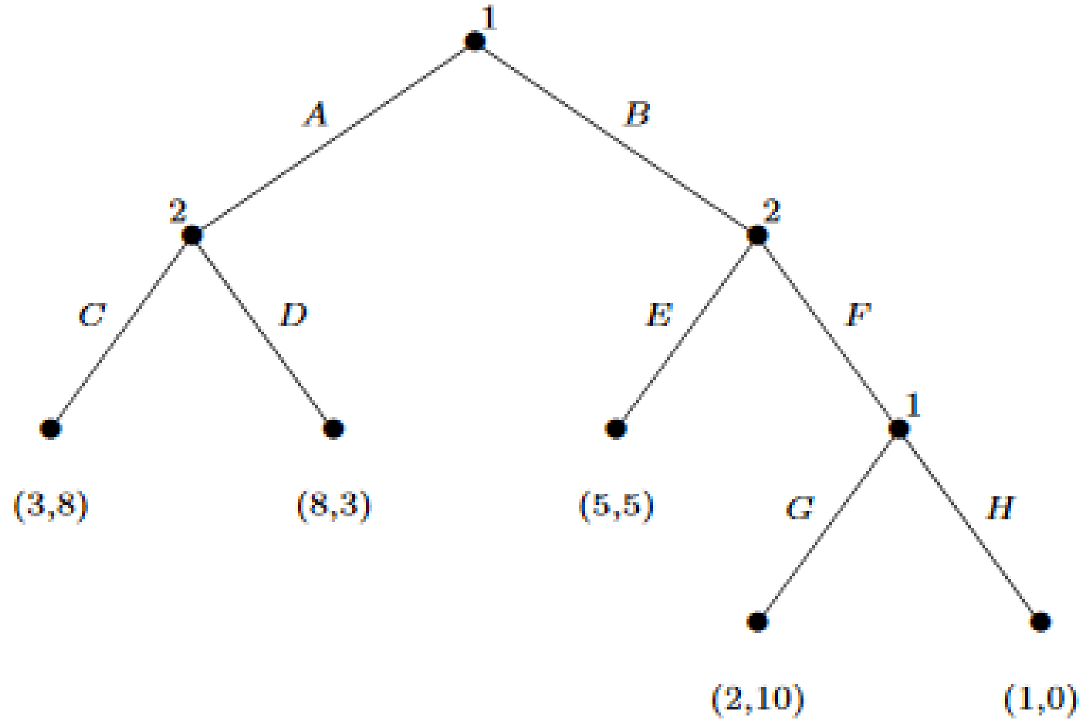
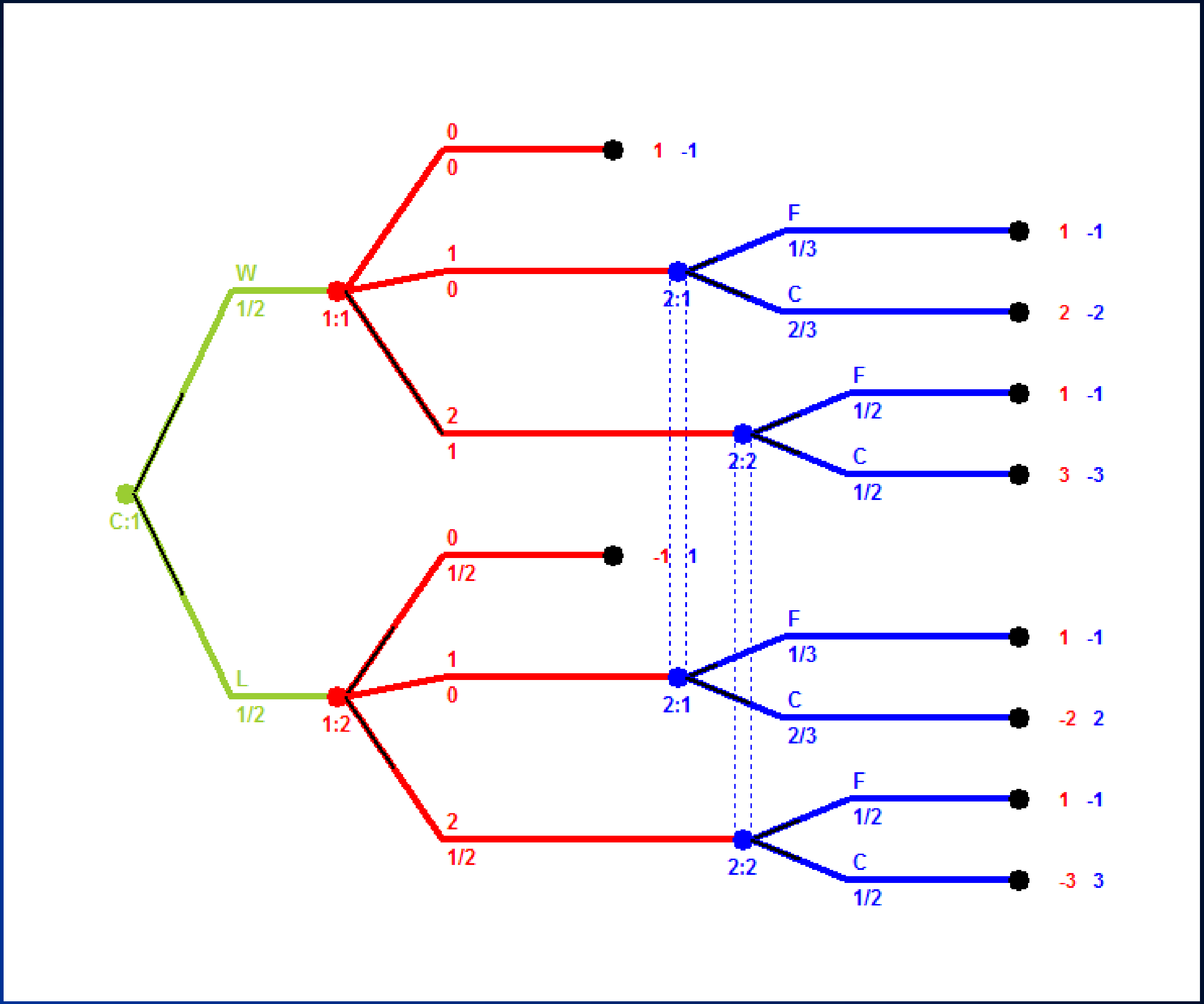


Figure 5.2: A perfect-information game in extensive form.

Cannot use backwards induction in games of imperfect information



Endgame definition

- E is an **endgame** of a game G if:

1. Set of E's nodes is a subset of set of G's nodes
2. If s' is a child of s in G and s is a node in E, then s' is also a node in E
3. If s is in the same information set as s' in G and s is a node in E, then s' is also a node in E

Can endgame solving guarantee equilibrium?

- Suppose that we computed an exact (full-game) equilibrium in the initial portion of the game tree prior to the endgame (the **trunk**), and computed an exact equilibrium in the endgame. Is the combined strategy an equilibrium of the full game?

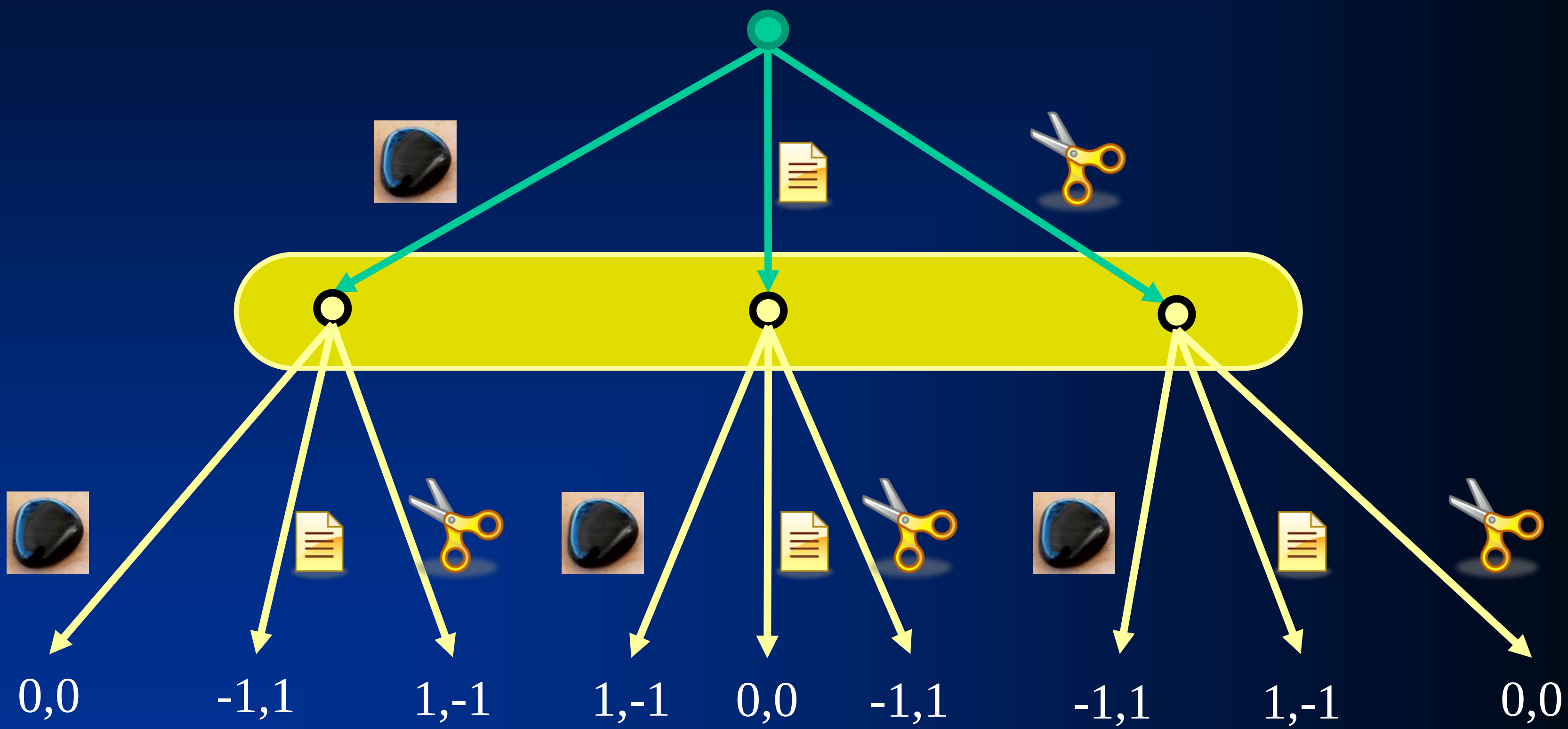
Can endgame solving guarantee equilibrium?

- No!
- Several possible reasons this may fail:
  - The game may have many equilibria, and we might choose one for the trunk that does not match up correctly with the one for the endgame
  - We may compute equilibria in different endgames that do not balance appropriately with each other

Can endgame solving guarantee equilibrium?

Proposition: There exist games with a unique equilibrium and a single endgame for which endgame solving can produce a non-equilibrium strategy profile in the full game

Can endgame solving guarantee equilibrium?



Endgame solving out of necessity for limit Texas hold ‘em agents 2003-2007

- The agent GS1 precomputed strategies only for the first two rounds, using rough approximations for the payoffs at the leaves of that trunk based on the (unrealistic) assumption that there was no betting in future rounds. Then in real time, the relevant endgame consisting of the final two rounds was solved using the LP algorithm [Gilpin-Sandholm AAAI ‘06].
- GS2 precomputed strategies for the first three rounds, using simulations to estimate the payoffs at the leaves; it then solved the endgames for the final two rounds in real time [Gilpin-Sandholm AAMAS ’07].
- At the time, these agents did not have scalable algorithms or resources for capability to solve abstractions of full four-round game at once, so a version of “endgame solving” was needed.

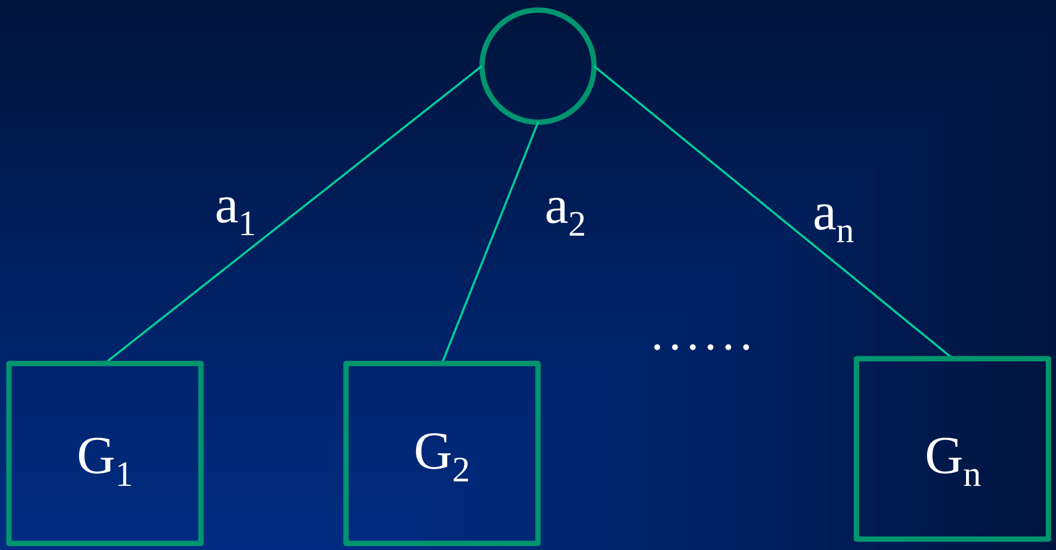
- 2007 – 2013, everyone: “Endgame solving is fundamentally flawed!!!”
  - “No point in doing it when we have the resources available to solve a reasonable abstraction of the full game already!”

Benefits of endgame solving

- Computation of exact (rather than approximate) equilibrium strategies in the endgames
- Computation of equilibrium refinements (e.g., undominated and ε-quasi-perfect equilibrium)
- Better abstractions in the endgame that is reached
  - Finer-grained abstractions
  - History-aware abstractions
  - Strategy-biased abstractions
- Solving the “off-tree problem”

- Endgame solving had not been implemented by any competitive agents for no-limit Texas hold 'em.
- Prior approaches assume that the private hand distributions leading into the endgame are independent, while they are actually dependent and the full joint distribution should be computed.
- The prior approaches use a single perfect-recall card abstraction that has been precomputed offline (which assumes a uniform random distribution for the opponent's hand distributions). In contrast, we use an imperfect-recall card abstraction that is computed in real time in a finer granularity than the initial offline abstraction and that is tailored specifically to the relevant distribution of the opponent's hands at the given hand history.
- Furthermore, the prior approaches did not compare performance between endgame solving and not using it (since the base strategies were not computed for the endgames), while we provide such a comparison.

Is there any hope?



Player 1 selects his action a_i, then the players play imperfect-information game G_i.

Is there any hope?

- Endgame solving produces strategies with low exploitability in games where the endgame is a significant strategic portion of the full game.
  - i.e., in games where any endgame strategy with high full-game exploitability can be exploited by the opponent by modifying his strategy just within the endgame.

Is there any hope?

- Proposition: If every strategy that has exploitability strictly more than ε in the full game has exploitability of strictly more than δ within the endgame, then the strategy output by a solver that computes a δ-equilibrium in the endgame induced by a trunk strategy t would constitute an ε-equilibrium of the full game when paired with t.

Endgame property

- We can classify different games according to property described by premise of proposition
  - If premise is satisfied, then we can say game satisfies the (ε, δ)-endgame property
- Interesting quantity would be smallest value $\varepsilon^*(\delta)$ such that game satisfies the (ε, δ)-endgame property for a given δ.
  - Game above has $\varepsilon^*(\delta) = \delta$ for each $\delta >= 0$
  - RPS has $\varepsilon^*(\delta) = 1$ for each $\delta >= 0$

Efficient algorithm for endgame-solving in large imperfect-information games

- **First step**: compute joint input distribution of private information using Bayes' rule
- Naïve approach requires iterating over all possible private hand combinations and for each pair looking up probability base strategy would take given sequence
  - This requires $O(n^2)$ lookups to the strategy table, where n is the number of possible hands (n = 1081 for river in poker)
  - Becomes bottleneck and makes real-time endgame solving computationally infeasible (takes > 1 min/hand)
  - Our algorithm uses just O(n) strategy table lookups (takes a few seconds per hand)

Algorithm for computing joint private hand distributions

- In short, the algorithm first computes distributions separately for each player (as done by the independent approach), then multiplies the probabilities together for hands that do not share a common card (and sets the joint probability to zero otherwise).
  - Utilizes indexing schemes to compute private card hand index and 7-card board
  - Maps 7-card index to 2-card index in main loop so we can determine which hands share a common card
  - Final loop is $O(n^2)$, though main bottleneck loop is $O(n)$

Main endgame-solving algorithm

- First compute joint hand distribution D[][]
- Next compute arrays E_1, E_2 of ***equities*** for each hand against opponent's distribution
  - For P1, do this by adding $D[h_1][h_2]$ to $E_1[h_1]$ for each hand h_2 with lower rank than that of h_1, and adding $D[h_1][h_2]/2$ for each hand with equal rank
- Compute information (card) abstractions A_1, A_2 by clustering elements of E_i into k_i buckets
  - $k_i = \text{floor}(T/b_i)$, where T is parameter and b_i number of action sequences in action abstraction
- Solve new abstracted game using an algorithm for computing an equilibrium (or a refinement)

Algorithm for computing endgame information abstractions

- Most prior work in poker uses k-means
  - However, this can be problematic. Suppose there are many hands with equity 0.7643, and also many hands with equity 0.7641. Then k-means would likely create separate clusters for these, and group hands with very different equities (e.g., 0.2 and 0.3) together if few hands have those equities.
- Instead, we use percentile hand strength
  - Break up the interval [0,1] into k_i regions of equal length
  - Group hand h_i into bucket floor($E_i[h_i] / k_i$)
  - We use modification where we first add a special bucket just for hands with $E_i[h_i] >= \text{alpha}$
  - Can result in significantly fewer than k_i buckets, since may be zero hands with E_i within some intervals. We take this into account, and reduce number of buckets in the card abstraction accordingly before solving the endgame.
  - The abstractions may be very different for the two players (and have different numbers of buckets)

Solving the abstracted endgame

- Solve the endgame with precomputed betting abstractions and computed information abstractions by applying an equilibrium-finding algorithm
  - If endgame has 10^8 or fewer states, can solve it exactly using LP, as opposed to using an iterative algorithm like CFR that is only guaranteed to approach equilibrium in the limit
  - Can also apply algorithms that compute certain refinements
- While card abstractions are computed independently, we use the joint distribution for determining probabilities that players are dealt hands from respective buckets when constructing the endgame

Experiments

- We tested our algorithm against the two strongest agents for two-player no-limit Texas Hold'em from the 2013 poker competition. The base agent was a version the agent we submitted to the 2014 ACPC from shortly before the competition.

- Endgame solver averaged around 8 seconds per hand (using Gurobi's LP solver)

Variance reduction

- Proposition: Let A_1 and A_2 be two algorithms that differ in play only for endgames. Then the difference in performance looking at only the hands where both make it to the same endgame is not an unbiased estimator of the overall performance difference.

- Proposition: Let A_1 and A_2 be two algorithms that differ in play only for endgames. Then the difference in performance looking at only the hands where both make it to some (but not necessarily the same) endgame is an unbiased estimator of the overall performance difference.

Experiments

- Statistically significant performance improvement against both opponents: 87+-50 vs. Hyperborean and 29+-25 vs. Slumbot
  - Results are from 100 duplicate matches against Hyperborean and 155 against Slumbot. Since each match is 3000 hands, this corresponds to 600,000/930,000 hands.
  - Out of these hands, both versions of our agent made it to the river round on 173,568 hands against Hyperborean and on 318,700 hands against Slumbot.
  - Results are from hands where both versions made it to the river, using our variance-reduction technique (would not have obtained statistical significance using prior duplicate approach).

Experiments

- Base agent used purification for all actions except first preflop action
  - Was shown to be best in prior experiments and was our standard setting for evaluating our base agent
- Endgame solver assumed that both agents used no thresholding when creating the input distributions
- Endgame solver did not do any rounding for the river

Problematic hands

1. We had A4s and folded preflop after putting in over half of our stack (human had 99).
   - We only need to win 25% of time against opponent's distribution for call to be profitable (we win 33% of time against 99).
   - Translation mapped opponent's raise to smaller size, which caused us to look up strategy computed thinking that pot size was much smaller than it was (7,000 vs. 10,000)
2. We had KT and folded to an all-in bet on turn after putting in ¾ of our stack despite having top pair and a flush draw
   - Human raised slightly below smallest size in our abstraction and we interpreted it as a call
   - Both 1 and 2 due to "off-tree problem"
3. Large all-in bet of 19,000 into small pot of 1700 on river without "blocker"
   - E.g., 3s2c better all-in bluff hand than 3c2c on JsTs4sKcQh
   - Endgame information abstraction algorithm doesn't fully account for "card removal"

Reflections on the First Man vs. Machine No-Limit Texas Hold ‘em Competition [Sigecom Exchanges ‘15, AI Magazine Summer ‘17]

- Two most important avenues for improvement
  - Solving the “off-tree problem”
  - Improved approach for information abstraction that better accounts for card removal/“blockers”
- Improved theoretical understanding of endgame solving
  - Works very well in practice despite lack of guarantees
  - Newer decomposition approach with guarantees does worse
- Bridge abstraction gap
  - Approaches with guarantees only scale to small games
  - Larger abstractions work well despite theoretical “pathologies”
- Diverse applications of equilibrium computation
- Theoretical questions for action translation/post-processing

Two most important avenues for improvement

- “The first is to develop an improved approach for the ``off-tree'' problem where we make a mistake due to a misperception of the actual size of the pot after translating an action for the opponent that is not in our action abstraction. We have outlined promising agendas for attacking this problem, including improved action abstraction and translation algorithms, **novel approaches for real-time computation that address the portion of the game prior to the final round**, and entirely new approaches specifically geared at solving the off-tree problem independently of the other problems.”

Two most important avenues for improvement

- “And the second is to develop an improved approach for information abstraction that better accounts for card removal/``blockers" (i.e., that accounts for the fact that us having certain cards in our hand modifies the probability of the opponent having certain hands). **This issue is most problematic within the information abstraction algorithm for the endgame, where the card removal effect is most significant due to the distributions for us and the opponent being the most well defined** (i.e., there is no more potential remaining in the hand due to uncertainty of public cards, and this relative certainty will likely cause the distributions to put positive weight on fewer hands), and it limits our ability to utilize large bet sizes, which have been demonstrated to be optimal in certain settings.”

- New game decomposition approach (CFR-d) has theoretical guarantee but performs worse empirically
  – Burch et al. AAAI-14

- Recent approach for safer endgame solving that maximizes the "endgame margin"
  – Moravic et al. AAAI-16

- Doug Polk related to me in personal communication after the competition that he thought the river strategy of Claudico using the endgame solver was the strongest part of the agent.

Second Brains vs. AI Competition

- Libratus: +14.7 BB/100 over 120,000 hands ($200k in prizes)
  - Claudico -9.16 BB/100 over 80,000 hands ($100k in prizes)

1. Libratus: 20-25 million core hours on supercomputer
   - Claudico: 2-3 million core hours on supercomputer

2. Improved equilibrium-finding algorithm “Regret-based pruning” which prunes actions with high regret early on in CFR so that the computation can eliminate large portions of the game tree following these “bad” actions.
   – Brown and Sandholm, “**Reduced Space and Faster Convergence in Imperfect-Information Games via Regret-Based Pruning**,” 2017 AAAI Workshop on Computer Poker and Imperfect Information

3. Improved endgame solver. Used supercomputer resources in real time. Was able to solve full turn and river endgames within around 20 seconds. Estimated that it would take 10+ minutes on normal machine.
   - Brown and Sandholm, "**Safe and Nested Endgame Solving in Imperfect-Information Games**," 2017 AAAI Workshop on Computer Poker and Imperfect Information
   - Used CFR instead of LP for endgame solving (to better capitalize on parallelization).

4. “Strategy improver”
   – “That's why Brown and Sandholm built a third system. Each evening, Brown would run an algorithm that could identify those patterns and remove them. "It could compute this overnight and have everything in place the next day," he says.”

5. Claudico’s mistakes → Libratus’ strengths
    – e.g., card removal/“blockers” and off-tree problem

DeepStack

- DeepStack agent from Alberta beat human “professionals” but not two-player no-limit Texas hold ‘em specialists in 2016 for 49 bb/100.
  - Libratus: +14.7 BB/100
  - Claudico: -9.16 BB/100
- It played 3000 hands per match against each human, against ~35 humans. Used variance reduction techniques for statistical significance.
- Published in “Science,” 2017.

- Michael Bowling, “DeepStack is ALL Endgame Solving!”
- Solves each round independently, assuming payoffs that were trained using deep learning.
- DeepStack acts very quickly in real time, but requires ~175 core years for the training, which is equivalent to several hundred computers for several months.

	Small Game	Large Game
Base Strategy	9.128	4.141
Unsafe	0.5514	39.68
Resolve	8.120	3.626
Maxmargin	0.9362	0.6121
Reach-Maxmargin	0.8262	0.5496

Table 1: Exploitability (evaluated in the game with no information abstraction) of the endgame-solving techniques.

Midgame solving
(Ganzfried & Hu ICTAI ‘17)

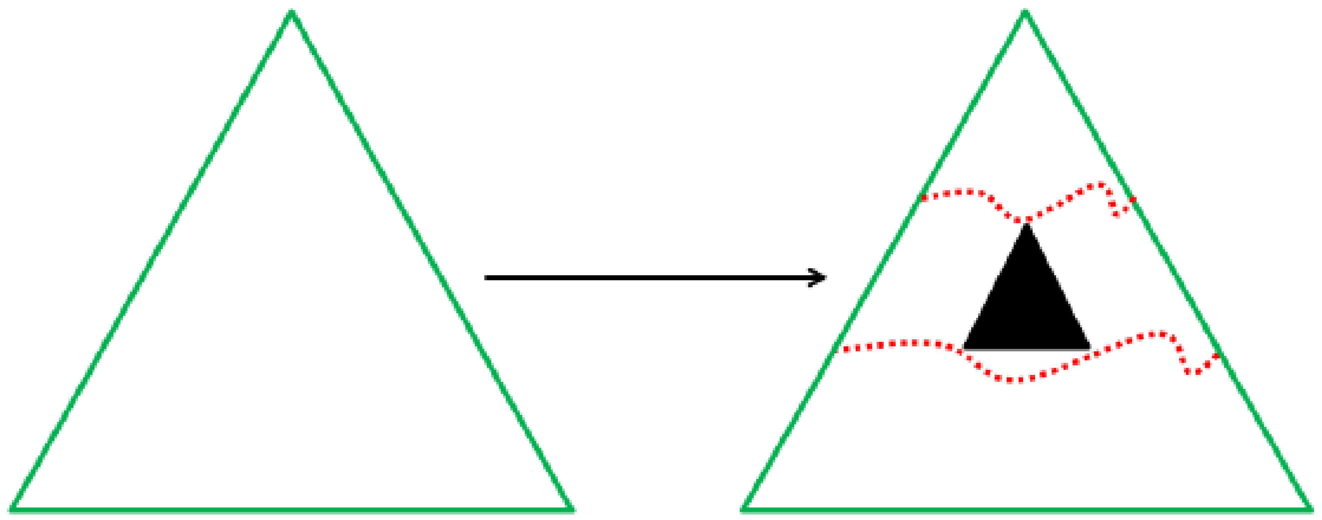


Fig. 3. *Midgame solving* discards the strategies that were precomputed for the midgames, then (re-)solves the relevant midgame that we have actually reached in real time to a greater degree of accuracy than in the offline computation, assuming a given payoff mapping for midgame terminal nodes.

Computing Nash equilibria in games with more than two players

- CFR can be run, but no significant guarantees other than that it cannot play an iteratively strictly dominated action.
- Developed new algorithms for computing ε-equilibrium strategies in multiplayer imperfect-information stochastic games
  - Models multiplayer poker tournament endgames [AAMAS08/IJCAI09]
- Most successful algorithm, called PI-FP, used a two-level iterative procedure
  - Outer loop is variant of policy iteration
  - Inner loop is an extension of fictitious play
- Proposition: If the sequence of strategies determined by iterations of PI-FP converges, then the final strategy profile is an equilibrium.
- We verified that our algorithms did in fact converge to ε-equilibrium strategies for very small ε.

The need for opponent exploitation

- Game-solving approach produces unexploitable (i.e., “safe”) strategies in two-player zero-sum games
- But it has no guarantees in general-sum and multiplayer games
- Furthermore, even in two-player zero-sum games, a much higher payoff is achievable against weak opponents by learning and exploiting their mistakes

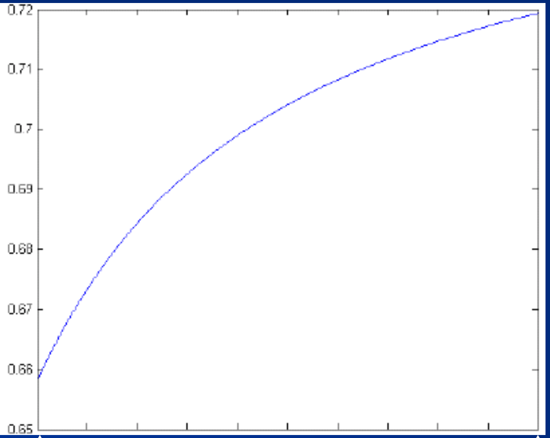
Opponent exploitation challenges

- Needs prohibitively many repetitions to learn in large games (only 3000 hands per match in the poker competition, so only have observations at a minuscule fraction of information sets)
- Partial observability of opponent's private information
- Often, there is no historical data on the specific opponent
  - Even if there is, it may be unlabelled or semi-labelled
- Recently, game-solving approach has significantly outperformed exploitation approaches in Texas hold 'em

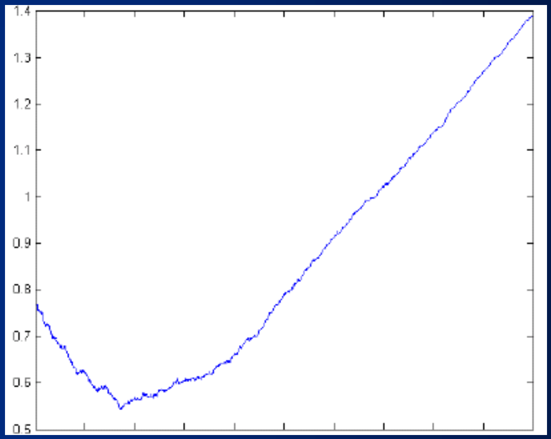
Experiments on opponent exploitation

- Significantly outperforms game-theory-based base strategy in 2-player limit Texas hold ‘em against
  - trivial opponents (e.g., one that always calls and one that plays randomly)
  - weak opponents from AAAI computer poker competitions
- Don’t have to turn this on against strong opponents

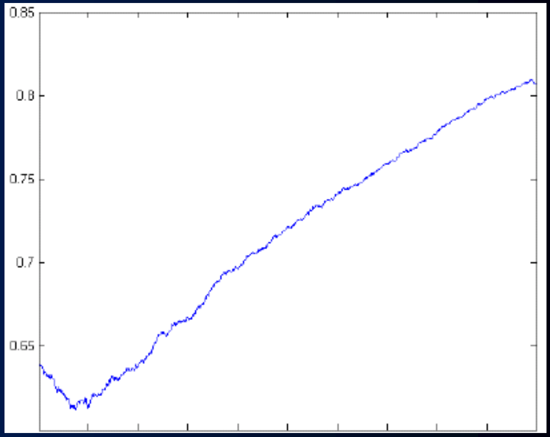
Opponent: ***Always fold***



Opponent: ***Always raise***



Opponent: ***GUS2***



Win rate

1,000

3,000

#hands

Safe opponent exploitation

- Definition. *Safe* strategy achieves at least the value of the (repeated) game in expectation

- Is safe exploitation possible (beyond selecting among equilibrium strategies in the one-shot game)?

Rock-Paper-Scissors

- Suppose the opponent has played Rock in each of the first 10 iterations, while we have played the equilibrium σ^*
- Can we exploit him by playing pure strategy Paper in the 11th iteration?
  - Yes, but this would not be safe!
- By similar reasoning, any deviation from σ^* will be unsafe
- So safe exploitation is not possible in Rock-Paper-Scissors

Rock-Paper-Scissors-Toaster

	rock	paper	scissors	toaster
Rock	0,0	-1, 1	1, -1	4, -4
Paper	1,-1	0, 0	-1,1	3, -3
Scissors	-1,1	1,-1	0,0	3, -3

- t is *strictly dominated*
  - s does strictly better than t regardless of P1’s strategy
- Suppose we play NE in the first round, and he plays t
  - Expected payoff of 10/3
- Then we can play R in the second round and guarantee at least 7/3 between the two rounds
- Safe exploitation is possible in RPST!
  - Because of presence of ‘gift’ strategy t

When can opponent be exploited safely?

- ~~Opponent played an (iterated weakly) dominated strategy?~~

R is a gift
but not iteratively weakly dominated

	L	M	R
U	3	2	10
D	2	3	0

- ~~Opponent played a strategy that isn't in the support of any eq?~~

R isn't in the support of any equilibrium
but is also not a gift

	L	R
U	0	0
D	-2	1

- **Definition.** We received a *gift* if opponent played a strategy such that we have an equilibrium strategy for which the opponent's strategy isn't a best response
- **Theorem.** Safe exploitation is possible iff the game has gifts

Exploitation algorithms

1. Risk what you've won so far
2. Risk what you've won so far in expectation (over nature's & own randomization), i.e., risk the gifts received
   – Assuming the opponent plays a nemesis in states we don't observe

- **Theorem.** A strategy for a two-player zero-sum game is safe iff it never risks more than the gifts received according to #2
- Can be used to make any opponent model / exploitation algorithm safe
- No prior (non-eq) opponent exploitation algorithms are safe
- We developed several new algorithms that are safe
  – Present analogous results and algorithms for extensive-form games of perfect and imperfect-information

Risk What You've Won in Expectation (RWYWE)

- Set $k^1 = 0$
- for $t = 1$ to T do
  - Set π^t_i to be k^t-safe best response to M
  - Play action a^t_i according to π^t_i
  - Update M with opponent's action a^t_{-i}
  - $\boxed{\text{Set } k^{t+1} = k^t + u_i(\pi^t_i, a_{-i}) - v^*}$

Game solving challenges

- Nash equilibrium lacks theoretical justification in certain game classes
  - E.g., games with more than two players
  - Even in two-player zero-sum games, certain refinements are preferable
- Computing Nash equilibrium is PPAD-complete in certain classes
- Even approximating NE in 2p zero-sum games very challenging in practice for many interesting games
  - Huge state spaces
- Robust exploitation is preferable

Major future avenue: Beyond two agents

- Many components of the standard and the endgame paradigms are directly applicable to more agents.
- Direct: abstraction, translation, post-processing, endgame/midgame solving, exploitation algorithms
  - Endgame solving: would require only two agents remaining in the endgame, which is common for postflop in poker. For more than two agents, would likely need CFR approximation.
- Equilibrium algorithms such as CFR can apply, but guarantees are limited.
  - Improved theoretical understanding of existing approaches.
  - New approaches with improved performance/theory.
  - New solution concepts altogether, with efficient algorithms for them.
- Safe exploitation, but guarantees maximin instead of value

www.computerpokercompetition.org/index.php/78-announcements/123-2018-call-for-participation

acpc 2018 poker

ANNUAL
Computer Poker Competition

Home About Competitions Downloads Forums Contact

2018 Call for Participation

The Annual Computer Poker Competition will be held again in February 2018. Noam Brown will be the incoming chair of the competition and Martin Schmid will be returning as the outgoing chairs. This time there will be a heads-up (two-player) no-limit Texas hold'em competition, and for the first time there will be a six-player no-limit Texas hold'em competition.

The plan is to use Amazon EC2 instances again this year, with a final submission deadline of December 20th 2017. Note that these machines are quite a bit less powerful than many desktop machines, so if computing resources are a significant issue for you, please check that this is adequate. The maximum submission size will remain the same at 250 GB this year for both the two-player and six-player competitions. The remaining technical specifications for the two-player competition remain the same as last year. The technical specifications for the six-player competition will be identical to the two-player competition, except that only 2 seconds per hand will be allowed rather than 7 (since each player will likely be involved in fewer hands). See http://aws.amazon.com/ec2/instance-types/ for details on Amazon instance types. It is possible to get access to a smaller testing instance through Amazon for free by creating an EC2 account, and all competitors will get access to a full sized competition machine for final testing at least one week prior to the final submission deadline.

Any individuals or teams that wish to participate should send an email by December 1, 2017 to chair@computerpokercompetition.org expressing their interest. If you require any special software - anything that is not installed by default on an Amazon EC2 instance - please let us know by December 1, so we can try to make sure your requirements are met. Your expression of interest email is not a formal commitment, but allows us to plan for the time and resources required to run the competition.