



PERSPECTIVES



Playing at a normal human pace, one can't beat a computer program for limit Texas Hold'em with statistical significance in a lifetime.

COMPUTER SCIENCE

Solving imperfect-information games

The smallest common poker game, two-player limit Texas Hold'em, is essentially solved

By **Tuomas Sandholm**

Imperfect-information games model settings where players have private information. Tremendous progress has been made in solving such games over the past 20 years, especially since the Annual Computer Poker Competition was established in 2006, where programs play each other. This progress can fuel the operationalization of seminal game-theoretic solution concepts into detailed game models, powering a host of applications in business (e.g., auctions and negotiations), medicine (e.g., making sophisticated sequential plans

against diseases), (cyber)security, and other domains. On page 145 of this issue, Bowling *et al.* (1) report on having computed a strategy for two-player limit Texas Hold'em poker that is so close to optimal that, at the pace a human plays poker, it cannot be beaten with statistical significance in a lifetime. While strong strategies have been computed for larger imperfect-information games as well (2–6), this is, to my knowledge, the largest imperfect-information game essentially solved to date, and the first one competitively played by humans that has now been essentially solved.

The general leading approach for solving imperfect-information games is shown in the figure; for a review, see (7). First, the game is abstracted to generate a smaller but

strategically similar game, reducing it to a size that can be tackled with an equilibrium-finding algorithm. Then, the abstract game is solved for equilibrium or near-equilibrium. A Nash equilibrium defines a notion of rational play. It is a profile of strategies, one per player, such that no player can increase her expected payoff by switching to a different strategy. A strategy for a player states for each information set where it is the player's turn, the probability with which the player should select each of her available actions. An information set is a collection of game states that cannot be distinguished by the player whose turn it is because of private information of the other players. Finally, the strategies from the abstract game are mapped back to the original game.

Carnegie Mellon University, Computer Science Department, Pittsburgh, PA, USA. E-mail: sandholm@cs.cmu.edu

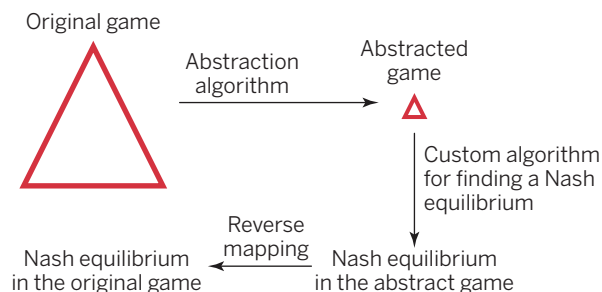
Two main kinds of abstraction are used. One is information abstraction, where it is assumed in the abstract game that a player does not know some information that she knows—that is, information sets are bundled. Lossless abstraction algorithms (8) yield an abstract game from which each equilibrium is also an equilibrium in the original game, and typically reduce the size of poker games by one-to-two orders of magnitude. My group (8) first used them to solve Rhode Island Hold'em, a benchmark introduced in 2001. Bowling *et al.* used lossless abstraction in the form of encoding the game in a way that avoids card suit symmetries. This reduced the number of information sets from 3.19×10^{14} to 1.38×10^{13} . Their scalable equilibrium-finding algorithm enabled them to essentially solve this lossless abstract game. For many larger games, the losslessly abstracted game would still be prohibitively large. Lossy abstraction algorithms (2, 3, 8–10) are used to create a smaller, coarser abstract game. An optimal strategy from such a game is typically not optimal for the original game.

The second method, action abstraction (2, 4, 5, 9), removes some actions from consideration in the game model, and is useful when the number of actions that a player can choose is large. For example, in two-player no-limit Texas Hold'em—a more popular game with 6.38×10^{61} information sets—a player can bet a variable number of chips, unlike in limit Texas Hold'em, where the bet size is fixed. Sophisticated techniques are then needed to map any action that the opponent plays that is not included in the abstract game to an action in the abstract game (6), because the opponent may try to manipulate the action abstraction—by clever bet-sizing in the case of poker.

Empirically, in Texas Hold'em, as a finer-grained abstraction is used, the quality of the strategy improves even when evaluated in the original game. Unfortunately, this is not always the case. Unlike in single-player settings, using a finer-grained abstraction can cause the computed strategy to be worse (more exploitable) in the original game. Despite this, some recent frameworks for lossy abstraction yield bounds on solution quality (5, 9). For a review of abstraction, see (10).

The second step of the game-solving process is finding an equilibrium in the abstract game (as shown in the figure). Before 2006, general-purpose linear programming solvers and the sequence-form representation

were used to solve small variants of poker or coarse abstractions of two-player limit Texas Hold'em. Since then, two families of dramatically more scalable equilibrium-finding algorithms and problem representations have been developed for two-player zero-sum games. One family is based on smoothed gradient descent algorithms and a decomposed problem representation (11, 12). The other family, counterfactual regret minimization (CFR) (13), is based on a form of self-play



Leading approach to imperfect-information games.

using no-regret learning, adapted cleverly so that regret updates can be computed at each information set separately, instead of the naïve approach that would require regrets to be updated for entire game strategies. The best available guarantees for CFR require $\sim 1/\epsilon^2$ iterations over the game tree to reach an ϵ -equilibrium, that is, strategies for the players such that no player can be exploited by more than ϵ by any strategy. The gradient-based algorithms require only $\sim 1/\epsilon$ (11) or $\sim \log(1/\epsilon)$ iterations (12). The latter approach matches the optimal number of iterations required (previously only achievable by interior-point methods that have prohibitive memory requirements). On the other hand, more effective sampling techniques have been developed for CFR than for the gradient-based algorithms, so quick approximate iterations can be used.

The key contribution of Bowling *et al.* is a scalable equilibrium-finding algorithm. It uses full CFR iterations rather than sampling. It distributes the computation across computers by dividing the game into disjoint pieces based on publicly observable information—public cards and past moves of the players—akin to related CFR work (6, 7). The numeric resolution of probabilities was reduced to save memory, with the drawback of slightly reducing solution quality. Careful compression techniques were used to store the game pieces on local disks and to bring them back into memory for updates. To empirically enhance speed, their algorithm never lets the regrets decrease below zero and uses the most recent computed strategies as the solution—instead of average strategies as in CFR convergence proofs.

Bowling *et al.* lower the bar for “solving” the game from finding an exact equilibrium or one within machine precision to requiring that at human playing speed, an adversary could not win with statistical significance in a lifetime. For many applications, this guarantee will be strong enough.

Much work is still required on tackling larger imperfect-information games (2–6). Another key direction is opponent exploitation—taking advantage of suboptimal play. Solely using machine learning for this typically opens oneself up to considerable exploitation in return, and one also typically accrues substantial losses during the learning process. Interesting recent approaches hybridize the game-theoretic approach and opponent exploitation in various ways (14, 15). It is even possible to exploit an opponent’s weaknesses more than any equilibrium strategy would, without opening oneself to any exploitation! This can be accomplished by—and only by—risking in one’s exploitation attempts winnings obtained so far via the opponent’s mistakes rather than luck (16).

Games with more than two players and that are not zero sum also deserve further attention. Most of the abstraction techniques apply here, but these equilibrium-finding problems are in a complexity class for which no polynomial-time algorithm is known. It is not even clear that finding a Nash equilibrium is the right goal in such games. Different equilibria can have different values to the players. If the opponents fail to play an equilibrium strategy, it may not be safe for us to play an equilibrium strategy. Finally, opponents may collude. ■

REFERENCES

1. M. Bowling *et al.*, *Science* **347**, 145 (2015).
2. E. Jackson, *Workshop on Computer Poker and Imperfect Information* (AAAI, Palo Alto, 2013), pp. 35–38.
3. N. Brown, S. Ganzfried, T. Sandholm, in *AAAI Workshop on Computer Poker and Imperfect Information* (AAAI, Palo Alto, 2015); www.cs.cmu.edu/~sandholm/2014champion.pdf
4. J. Hawkin, R. Holte, D. Szafron, in *Proc. 26th AAAI Conf. Artif. Intell.* (AAAI, Palo Alto, 2012), pp. 1924–1930.
5. N. Brown, T. Sandholm, in *Proc. 28th AAAI Conf. Artif. Intell.* (AAAI, Palo Alto, 2014), pp. 594–601.
6. S. Ganzfried, T. Sandholm, in *Proc. 23rd Int. Joint Conf. Artif. Intell.* (AAAI, Palo Alto, 2013), pp. 120–128.
7. T. Sandholm, *AI Mag.* **31**, 13 (2010).
8. A. Gilpin, T. Sandholm, *J. ACM* **54**, 25, es (2007).
9. C. Kroer, T. Sandholm, in *Proc. 15th ACM Conf. Econ. Comput.* (ACM, New York, 2014), pp. 621–638.
10. T. Sandholm, in *Proc AAAI Conf. Artif. Intell.* (AAAI, Palo Alto, 2015); www.cs.cmu.edu/~sandholm/abstraction.pdf
11. S. Hoda *et al.*, *Math. Oper. Res.* **35**, 494 (2010).
12. A. Gilpin *et al.*, *Math. Program.* **133**, 279 (2012).
13. M. Zinkevich, M. Johanson, M. Bowling, C. Piccione, *Adv. Neural Inf. Process. Syst.* **20**, 905 (2008).
14. O. S. Ganzfried, T. Sandholm, in *Proc. 10th Int. Conf. Autonomous Agents Multiagent Syst.* (IFAAMAS, Richland, SC, 2011), pp. 781–788.
15. N. Bard *et al.*, in *Proc. 12th Int. Conf. Autonomous Agents and Multiagent Systems* (IFAAMAS, Richland, SC, 2013), pp. 255–262.
16. S. Ganzfried, T. Sandholm, in *Proc. 13th Conf. on Electronic Commerce* (ACM, New York, 2012), pp. 587–604.

10.1126/science.aaa4614



Solving imperfect-information games

Tuomas Sandholm
Science **347**, 122 (2015);
DOI: 10.1126/science.aaa4614

This copy is for your personal, non-commercial use only.

If you wish to distribute this article to others, you can order high-quality copies for your colleagues, clients, or customers by [clicking here](#).

Permission to republish or repurpose articles or portions of articles can be obtained by following the guidelines [here](#).

The following resources related to this article are available online at www.sciencemag.org (this information is current as of January 8, 2015):

Updated information and services, including high-resolution figures, can be found in the online version of this article at:

<http://www.sciencemag.org/content/347/6218/122.full.html>

A list of selected additional articles on the Science Web sites **related to this article** can be found at:

<http://www.sciencemag.org/content/347/6218/122.full.html#related>

This article **cites 6 articles**, 1 of which can be accessed free:

<http://www.sciencemag.org/content/347/6218/122.full.html#ref-list-1>

This article appears in the following **subject collections**:

Computers, Mathematics

http://www.sciencemag.org/cgi/collection/comp_math