

## COMPUTER SCIENCE

# Mastering board games

A single algorithm can learn to play three hard board games

By Murray Campbell

From the earliest days of the computer era, games have been considered important vehicles for research in artificial intelligence (AI) (1). Game environments simplify many aspects of real-world problems yet retain sufficient complexity to challenge humans and machines alike. Most programs for playing classic board games have been largely human-engineered (2, 3). Sophisticated search methods, complex evaluation functions, and a variety of game-specific tricks have allowed programs to surpass the best human players. More recently, a learning approach achieved superhuman performance in the hardest of the classic games, Go (4), but was specific for this game and took advantage of human-derived game-specific knowledge. Subsequent work (5) removed the need for human knowledge, and additional algorithmic enhancements delivered further performance improvements. On page 1140 of this issue, Silver *et al.* (6) show that a generalization of this approach is effective across a variety of games. Their AlphaZero system learned to play three challenging games (chess, shogi, and Go) at the highest levels of play seen.

AlphaZero is based on reinforcement learning (7), a very general paradigm for learning to act in an environment that rewards useful actions. In the case of board games, the learning agent plays moves in the game and is typically trained by playing large numbers of games against itself. The first major success for reinforcement learning and games was the TD-Gammon program (8), which learned to play world-class backgammon in the early 1990s by using neural networks. More recently, deep (many-layer) neural networks were combined with reinforcement learning in an approach dubbed “deep reinforcement learning,” which received widespread interest after it was successfully applied to learn Atari video games directly from screen input (9).

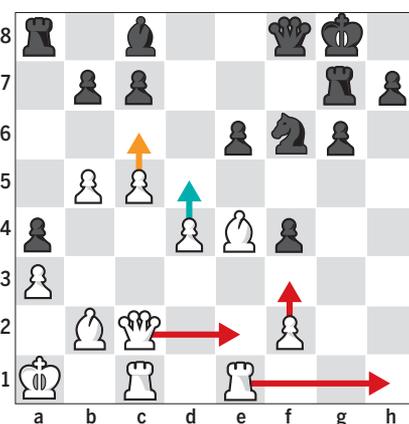
The approach described by Silver *et al.* augments deep reinforcement learning with a general-purpose searching method, Monte Carlo tree search (MCTS) (10). Although MCTS has been the standard searching method used in Go programs for some time, until now, there had been little evidence of its

value in chess or shogi programs. The strongest programs in both games have relied on variations of the alpha-beta algorithm, used in game-playing programs since the 1950s.

Silver *et al.* demonstrated the power of combining deep reinforcement learning with an MCTS algorithm to learn a variety of games from scratch. The training methodology used in AlphaZero is a slightly modified version of that used in the predecessor system AlphaGo Zero (5). Starting from randomly initialized parameters, the neural network continually updates the parameters on

## Contemplating the next move

In the game between AlphaZero (white) and Stockfish (black), there were several moves that were reasonable for AlphaZero to consider. After 1000 move-sequence simulations, the red moves were rejected, and after 100,000 simulations, AlphaZero chose the blue move over orange.



basis of the outcome of self-play games. AlphaZero learned to play each of the three board games very quickly by applying a large amount of processing power, 5000 tensor processing units (TPUs), equivalent to a very large supercomputer.

Once trained, evaluating the systems is not entirely trivial, and there are many pitfalls that can affect the measurements. Silver *et al.* used a large variety of testing conditions which, taken together, provide convincing evidence of the superiority of the trained systems over the previous state-of-the-art programs. Some of the early test games played between AlphaZero and the chess program Stockfish were released to the public and created something of a sensation in the chess

community, with much analysis and commentary on the amazing style of play that AlphaZero exhibited (see the figure). Note that neither the chess or shogi programs could take advantage of the TPU hardware that AlphaZero has been designed to use, making head-to-head comparisons more difficult.

Chess, shogi, and Go are highly complex but have a number of characteristics that make them easier for AI systems. The game state is fully observable; all the information needed to make a move decision is visible to the players. Games with partial observability, such as poker, can be much more challenging, although there have been notable successes in games like heads-up no-limit poker (11, 12). Board games are also easy in other important dimensions. For example, they are two-player, zero-sum, deterministic, static, and discrete, all of which makes it easier to perfectly simulate the evolution of the game state through arbitrary sequences of moves. This ability to easily simulate future states makes MCTS, as used in AlphaZero, practical. Multiplayer video games such as StarCraft II (13) and Dota 2 (14) have been proposed as the next game-playing challenges as they are partially observable and have very large state spaces and action sets, creating problems for AlphaZero-like reinforcement learning approaches.

Games have been popular research domains in AI in part because it is easy to identify games in which humans are better than computers. Chess, shogi, and Go are immensely complex, and numerous human players have devoted much of their lives to understanding and playing these games at the professional level. The AlphaZero approach still has limitations that could be addressed (for example, large computational requirements, brittleness, and lack of interpretability), but this work has, in effect, closed a multidecade chapter in AI research. AI researchers need to look to a new generation of games to provide the next set of challenges. ■

## REFERENCES AND NOTES

1. C. Shannon, *Philos. Mag.* **41**, 256 (1950).
2. J. Schaeffer *et al.*, *AI Mag.* **17**, 21 (1996).
3. M. Campbell *et al.*, *Artif. Intell.* **134**, 57 (2002).
4. D. Silver *et al.*, *Nature* **529**, 484 (2016).
5. D. Silver *et al.*, *Nature* **550**, 354 (2017).
6. D. Silver *et al.*, *Science* **362**, 1140 (2018).
7. R. Sutton, A. Barto, *Reinforcement Learning: An Introduction* (MIT Press, 1998).
8. G. Tesauro, *Commun. ACM* **38**, 58 (1995).
9. V. Mnih *et al.*, *Nature* **518**, 529 (2015).
10. R. Coulom, *Int. Conf. Comput. Games* **2006**, 72 (2006).
11. M. Moravčík *et al.*, *Science* **356**, 508 (2017).
12. N. Brown, T. Sandholm, *Science* **359**, 418 (2018).
13. O. Vinyals *et al.*, arXiv:1708.04782 [cs.LG] (16 August 2017).
14. OpenAI, OpenAI Five, 25 June 2018; <https://blog.openai.com/openai-five/>.

## ACKNOWLEDGMENTS

Thanks to T. Klingner and G. Tesauro for their comments.

10.1126/science.aav1175

## Mastering board games

Murray Campbell

*Science* **362** (6419), 1118.  
DOI: 10.1126/science.aav1175

### ARTICLE TOOLS

<http://science.sciencemag.org/content/362/6419/1118>

### RELATED CONTENT

<http://science.sciencemag.org/content/sci/362/6419/1087.full>  
<http://science.sciencemag.org/content/sci/362/6419/1140.full>

### REFERENCES

This article cites 11 articles, 3 of which you can access for free  
<http://science.sciencemag.org/content/362/6419/1118#BIBL>

### PERMISSIONS

<http://www.sciencemag.org/help/reprints-and-permissions>

Use of this article is subject to the [Terms of Service](#)

---

*Science* (print ISSN 0036-8075; online ISSN 1095-9203) is published by the American Association for the Advancement of Science, 1200 New York Avenue NW, Washington, DC 20005. 2017 © The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works. The title *Science* is a registered trademark of AAAS.