

Special Issue on Monte Carlo Techniques and *Computer Go*

THE technique of Monte Carlo tree search (MCTS) has revolutionized the field of computer game playing, and is starting to have an impact in other search and optimization domains as well. In past decades, the dominant paradigm in game algorithms was alpha-beta search. This technique, with many refinements and game-specific engineering, led to breakthrough performances in classic board games such as *Chess*, *Checkers*, and *Othello*. After Deep Blue's famous victory over Kasparov in 1996, some of the research focus shifted to games where alpha-beta search was not sufficient. Most prominent among these games was the ancient Asian game of *Go*. Despite much effort, progress remained slow for another decade. During the last few years, the use of MCTS techniques in *Computer Go* has really taken off, but the groundwork was laid much earlier. In 1990, Abramson [1] proposed to model the expected outcome of a game by averaging the results of many random games. In 1993, Brüggmann [2] proposed Monte Carlo techniques for *Go* using almost random games, and developed the refinement he termed all-moves-as-first (AMAF). Ten years later, a group of French researchers working with Bouzy and Cazenave took up the idea [3]. Bouzy's Indigo program used Monte Carlo simulation to decide between the top moves proposed by a classical knowledge-based *Go* engine.

Coulom's Crazy Stone [4] was the first to add the crucial second element, a selective game tree search controlled by the results of the simulations. The last piece of the puzzle was the upper confidence tree (UCT) algorithm of Kocsis and Szepesvari [5], which applied ideas from the theory of multiarmed bandits to the problem of how to selectively grow a game tree. Gelly and Wang developed the first version of MoGo [6], which among other innovations combined Coulom's ideas, the UCT algorithm, and pattern-directed simulations. AMAF was revived and extended in Gelly and Silver's rapid action value estimate (RAVE), which computes AMAF statistics in all nodes of the UCT tree. Rapid progress in applying knowledge and parallelizing the search followed. Today, programs such as MoGo/MoGoTW, Crazy Stone, Fuego, Many Faces of Go, and Zen have achieved a level of play that seemed unthinkable only a decade ago. These programs are now competitive at a professional level for 9×9 *Go* and amateur Dan strength on 19×19 [7].

One measure of success is competitions. In *Go*, Monte Carlo programs now completely dominate classical programs on all board sizes (though no one has tried boards larger than 19×19). Monte Carlo programs have achieved considerable success in play against humans. An early sign of things to come was a series of games on a 7×7 board between Crazy Stone and pro-

fessional 5th Dan Guo Juan. Crazy Stone demonstrated almost perfect play. In 2009, a series of matches held on a 9×9 board, culminated in program wins playing as both white (the easier color) with Fuego and black with MoGo/MoGoTW against the top level professional *Go* player Chun-Hsun Chou. In 2010, MoGo and Many Faces of Go achieved wins against strong amateur players on 13×13 with only two handicap stones. On the full 19×19 board, programs have racked up a number of wins (but still a lot more losses) on six and seven handicap stones against top professional *Go* players [8], [9].

Besides rapid progress in *Go*, the most exciting recent developments in MCTS have shown an ever increasing array of applications. In games such as *Hex*, *Havannah*, and *Lines of Action*, MCTS is the state of the art. MCTS can play very well even with little knowledge about the game as evidenced by its success in general game playing. In areas as diverse as energy optimization problems, tuning of libraries, domain-independent planning, and solving Markov decision processes (MDPs), techniques inspired by MCTS are rapidly being developed and applied. However, current MCTS techniques do not work well for all games or all search problems. This poses some interesting questions. When and why does it succeed and fail? How can it be extended to new applications where it does not work yet? How best may it be combined with other approaches such as classical minimax search and knowledge-based methods?

The purpose of this Special Issue on Monte Carlo Techniques and *Computer Go* is to publish high-quality papers reporting the latest research covering the theory and practice of these and other methods applied to *Go* and other games. The special issue received eighteen paper submissions, of which eight have been accepted. These papers cover *Go*, *Lines of Action*, *Hex*, single-player general game playing, parallelization in *Go*, and analyzing game records using on Monte Carlo techniques.

The first paper, "Current frontiers in *Computer Go*" by Rimmel *et al.*, presents an overview of the state of the art in *Computer Go* by some of the current members of the MoGo project, shows the many similarities and the rare differences between the current best programs, and reports the results of the *Computer Go* event organized at the 2009 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2009). Importantly, the first ever win of a computer against a 9th Dan professional player in 9×9 *Go* occurred at this event.

The second paper, "Monte Carlo tree search in *Lines of Action*" of Winands *et al.*, presents a MCTS-based program for playing the game *Lines of Action* (LOA). With the improved MCTS variant, the proposed program is able to outperform even the world's strongest alpha-beta-based LOA program. This is an important milestone for MCTS because the traditional game-

tree search approach has been considered to be the better suited for playing LOA.

The third paper, “Monte Carlo tree search in *Hex*,” by Arneson *et al.*, describes MoHex, the MCTS *Hex* player that won gold in the 2009 Computer Olympiad. The main contributions to MCTS include using inferior cell analysis and connection strategy computation to prune the search tree. In particular, the authors run their random game simulations not on the actual game position, but on a reduced equivalent board.

The fourth paper, “FUEGO—An open-source framework for board games and *Go* engine based on Monte Carlo tree search” by Enzenberger *et al.*, gives an overview of the development and current state of the FUEGO project and describes the reusable components of the software framework and specific algorithms used in the *Go* engine. FUEGO was the first program to win a game against a top professional player in 9×9 *Go*.

The fifth paper, “Combining UCT and nested Monte Carlo search for single-player general game playing” by Méhat and Cazenave, compares nested Monte Carlo search (NMC), upper confidence bounds for trees (UCT-T), UCT with transposition tables (UCT+T), and a simple combination of NMC and UCT+T (MAX) on single-player games of the past General Game Playing (GGP) competitions. The experimental results show that the transposition tables improve UCT and that MAX is the best of these four algorithms. Using UCT+T, the program Ary won the 2009 GGP competition.

The sixth paper, “Evaluating root parallelization in *Go*” by Soejima *et al.*, discusses the various parallelization methods proposed for *Computer Go*. The authors analyze the performance of two root parallelization methods: the standard strategy based on average selection and the proposed strategy based on majority voting. The proposed algorithms are simple and generic and can be considered far from *Go*. The experimental results with 64 central processing unit (CPU) cores show that majority voting outperforms average selection.

The seventh paper, “Evaluation of game tree search methods by game records” by Takeuchi and Kaneko, presents a method of evaluating game tree search methods, including standard min–max search and Monte Carlo tree search. The authors applied the proposed method to *Go*, *Shogi*, and *Chess*, and by comparing the results with empirical understanding of the performance of various game tree search methods and with the results of self-play shows that the proposed method is efficient and effective.

The last paper, “The power of forgetting: Improving the last-good-reply policy in Monte Carlo *Go*” by Baier and Drake, describes an improvement to Drake’s last-good-reply policy, which is a method for online policy improvement that favors recently successful replies to the opponent’s last move. In their paper, Baier and Drake show that forgetting these replies as soon as they fail improves the playing strength of their *Go* program Orego. Surprisingly, remembering the win rate of every reply is not as effective as simply remembering the last good reply.

As guest editors of this special issue, we thank the authors for their contributions. We also would like to thank M.-H. Wang and

L.-W. Wu, members of the Ontology Application and Software Engineering (OASE) Laboratory at the National University of Tainan (NUTN), Taiwan, for their support of this Special Issue. We are most grateful to the referees for spending their valuable time in reviewing the manuscripts and providing kind cooperation and help. Finally, we greatly appreciate Prof. S. Lucas (Editor-in-Chief) and S. Woollam (Editorial Assistant) of the IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES for providing us with the opportunity to edit and publish this Special Issue, as well as for their valuable instructions in the editorial process.

CHANG-SHING LEE, *Guest Editor*
Department of Computer Science
and Information Engineering
National University of Tainan
Tainan, Taiwan
leecs@mail.nutn.edu.tw

MARTIN MÜLLER, *Guest Editor*
Department of Computing Science
University of Alberta
Edmonton, ABCanada
mmueller@ualberta.ca

OLIVIER TEYTAUD, *Guest Editor*
Thème Apprentissage et
Optimisation (TAO)
INRIA-Saclay
Paris, France
olivier.teytaud@inria.fr

REFERENCES

- [1] B. Abramson, “Expected-outcome: A general model of static evaluation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 2, pp. 182–193, Feb. 1990.
- [2] B. Brüggmann, “Monte Carlo *Go*,” 1993 [Online]. Available: <http://www.ideaest.com/vegos/MonteCarloGo.pdf>
- [3] B. Bouzy and T. Cazenave, “Computer *Go*: An AI-oriented survey,” *Artif. Intell. J.*, vol. 132, no. 1, pp. 39–103, 2001.
- [4] R. Coulom, “Efficient selectivity and backup operators in Monte-Carlo tree search,” in *Proc. 5th Int. Conf. Comput. Games*, Turin, Italy, 2006, pp. 72–83.
- [5] L. Kocsis and C. Szepesvari, “Bandit based Monte-Carlo planning,” in *Eur. Conf. Mach. Learn.*, 2006, vol. 4212, pp. 282–293.
- [6] Y. Wang and S. Gelly, “Modifications of UCT and sequence-like simulations for Monte-Carlo *Go*,” in *Proc. IEEE Symp. Comput. Intell. Games*, 2007, pp. 175–182.
- [7] C. S. Lee, M. H. Wang, C. Chaslot, J. B. Hoock, A. Rimmel, O. Teytaud, S. R. Tsai, S. C. Hsu, and T. P. Hong, “The computational intelligence of MoGo revealed in Taiwan’s computer *Go* tournaments,” *IEEE Trans. Comput. Intell. AI Games*, vol. 1, no. 1, pp. 73–89, Mar. 2009.
- [8] J. B. Hoock, C. S. Lee, A. Rimmel, F. Teytaud, M. H. Wang, and O. Teytaud, “Intelligent agents for the game of *Go*,” *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 28–42, Nov. 2010.
- [9] C. S. Lee, M. H. Wang, O. Teytaud, and Y. L. Wang, “The game of *Go* @ IEEE WCCI 2010,” *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 6–7, Nov. 2010.



Chang-Shing Lee (SM'09) received the Ph.D. degree in computer science and information engineering from the National Cheng Kung University, Tainan, Taiwan, in 1998.

Currently, he is a Professor at the Department of Computer Science and Information Engineering and Director of the Computer Center, National University of Tainan (NUTN), Tainan, Taiwan. His major research interests are in ontology applications, knowledge management, capability maturity model integration (CMMI), meeting scheduling, and artificial intelligence. He is also interested in intelligent agent, web services, fuzzy theory and application, genetic algorithm, and image processing. He also holds several patents on ontology engineering, document classification, image filtering, and healthcare.

Dr. Lee is the Emergent Technologies Technical Committee (ETTC) Chair of the IEEE Computational Intelligence Society (CIS) from 2009 to 2010, and was the ETTC Vice Chair of the IEEE CIS in 2008. He is the Committee Member of the IEEE CIS International Task Force on Intelligent Agents and on Emerging Technologies for Computer Go. Additionally, he is also the member of

the IEEE SMC Technical Committee on Intelligent Internet System (TCIIS). He also serves as an Associate Editor of the IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES and the *Journal of Ambient Intelligence & Humanized Computing* (AIHC), an Editorial Board member for the *Applied Intelligence*, the *Journal of Advanced Computational Intelligence and Intelligent Informatics* (JACIII), and *Open Cybernetics and Systemics Journal*, and a Guest Editor for the IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, the *Applied Intelligence Journal*, the *International Journal of Intelligent System* (IJIS), the *International Journal of Fuzzy Systems* (IJFS), and the *Journal of Internet Technology* (JIT). He is also the Program Committee member of more than 40 conferences. He is a member of the Taiwanese Association for Artificial Intelligence (TAAI) and Software Engineering Association Taiwan (SEAT).

(SM'09) received the Ph.D. degree in computer science and information engineering from the National Cheng Kung University, Tainan, Taiwan, in 1998.

Currently, he is a Professor at the Department of Computer Science and Information Engineering and Director of the Computer Center, National University of Tainan (NUTN), Tainan, Taiwan. His major research interests are in ontology applications, knowledge management, capability maturity model integration (CMMI), meeting scheduling, and artificial intelligence. He is also interested in intelligent agent, web services, fuzzy theory and application, genetic algorithm, and image processing. He also holds several patents on ontology engineering, document classification, image filtering, and healthcare.

Dr. Lee is the Emergent Technologies Technical Committee (ETTC) Chair of the IEEE Computational Intelligence Society (CIS) from 2009 to 2010, and was the ETTC Vice Chair of the IEEE CIS in 2008. He is the Committee Member of the IEEE CIS International Task Force on Intelligent Agents and on Emerging Technologies for Computer Go. Additionally, he is also the member of the IEEE SMC Technical Committee on Intelligent Internet System (TCIIS). He also serves as an Associate Editor of the IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES and the *Journal of Ambient Intelligence & Humanized Computing* (AIHC), an Editorial Board member for the *Applied Intelligence*, the *Journal of Advanced Computational Intelligence and Intelligent Informatics* (JACIII), and *Open Cybernetics and Systemics Journal*, and a Guest Editor for the IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, the *Applied Intelligence Journal*, the *International Journal of Intelligent System* (IJIS), the *International Journal of Fuzzy Systems* (IJFS), and the *Journal of Internet Technology* (JIT). He is also the Program Committee member of more than 40 conferences. He is a member of the Taiwanese Association for Artificial Intelligence (TAAI) and Software Engineering Association Taiwan (SEAT).



Martin Müller received the Ph.D. degree in computer science from ETH Zürich, Switzerland, in 1995.

Currently, he is a Professor at the Department of Computing Science, University of Alberta, Edmonton, AB, Canada. He is the leader of the FUEGO project and also acts as the *Go* expert. His research interests are in search algorithms, with applications to *Computer Go* and planning.



Olivier Teytaud was born in 1975. He received the M.S. degree in computer science from the University of Normale Sup, Lyon, France, in 1998 and the Ph.D. degree from the Lyon 2 University, Lyon, France, in 2001.

Currently, he is a Researcher at the Thème Apprentissage et Optimisation (TAO), Inria Saclay-IDF, Cnrs, Lri, University Paris-Sud, Paris, France. He works in artificial intelligence, statistical learning, evolutionary algorithms, and games.