Preface

This book contains the papers of the 9th Computer and Games Conference (CG 2016) held in Leiden, The Netherlands. The conference took place from June 29 to July 1, 2016, in conjunction with the 19th Computer Olympiad and the 22nd World Computer-Chess Championship.

The Computers and Games Conference series is a major international forum for researchers and developers interested in all aspects of artificial intelligence and computer game playing. During the Leiden conference, a Workshop on Neural Networks in Games was organized; the exciting results on Go from 2015–2016 were in everybody's mind. Moreover, there was an invited talk by Aja Huang (Google DeepMind) on Alpha Go, titled "Alpha Go: Combining Deep Neural Networks with Tree Search." Earlier conferences took place in Tsukuba (1998), Hamamatsu (2000), Edmonton (2002), Ramat-Gan (2004), Turin (2006), Beijing (2008), Kanazawa (2010), and Yokohama (2013).

The Program Committee (PC) was pleased to see that so much progress was made in new games and that new techniques were added to the recorded achievements. In this conference, 30 papers were submitted. Each paper was sent to at least three reviewers. If conflicting views on a paper were reported, the reviewers themselves arrived at a final decision. With the help of external reviewers (see after the preface), the PC accepted 20 papers for presentation at the conference and publication in these proceedings. As usual we informed the authors that they submitted their contribution to a post-conference editing process. The two-step process is meant (a) to give authors the opportunity to include the results of the fruitful discussion after the lecture in their paper, and (b) to maintain the high-quality level of the CG series. The authors enjoyed this procedure.

The aforementioned set of 20 papers covers a wide range of computer games and many different research topics. We grouped the topics into the following four main classes, which determined the order of publication: Monte Carlo Tree Search (MCTS) and its enhancements (seven papers), concrete games (seven papers), theoretical aspects and complexity (five papers), and cognition model (one paper). The paper "Using Partial Tablebases in Breakthrough" by Andrew Isaac and Richard Lorentz received the Best Paper Award.

We are sure that the readers will enjoy the research efforts presented by the authors. Below, we introduce them in the topics investigated by brief characterizations of the papers largely paraphrased by ideas as submitted by the authors, in particular in the abstract. The aim is to show a connection between the contributions and to provide insights into the research progress.

Monte Carlo Tree Search

The seven topics discussed in the area of MCTS are as follows (the game area is mentioned in brackets); Partial Tablebases (Breakthrough), Deep Convolutional Neural Network (Go), Parameterized Poker Squares (Poker), Robust Exploration (Go), Pruning Playouts (Havannah), Fast Seed Learning (Go), and Heuristic Function Evaluation Framework (several games).

"Using Partial Tablebases in Breakthrough" is written by Andrew Isaac and Richard Lorentz. In the game of Breakthrough the endgame is reached when there are still many pieces on the board. This means that there are too many possible positions to be able to construct a reasonable endgame tablebase on the standard 8×8 board, or even on a 6×6 board. The fact that Breakthrough pieces only move forward allows researchers to create partial tablebases on the last n rows of each side of the board. The authors show how this construction results in a much stronger MCTS-based 6×6 player and even allows positions to be solved that would otherwise be out of reach.

"Using Deep Convolutional Neural Networks in Monte Carlo Tree Search" is authored by Tobias Graf and Marco Platzner. Deep Convolutional Neural Networks have revolutionized Computer Go. Large networks have emerged as state-of-the-art models for move prediction and are used not only as stand-alone players but also inside MCTS to select and bias moves. Using neural networks inside the tree search is a challenge due to their slow execution time even if accelerated on a GPU. In this paper the authors evaluate several strategies to limit the number of nodes in the search tree in which neural networks are used. All strategies are assessed using the freely available cuDNN library. The authors compare the strategies against an optimal upper bound that can be estimated by removing timing constraints. They show that the best strategies are only 50 ELO points worse than this upper bound.

"Monte Carlo Approaches to Parameterized Poker Squares" is written by Todd Neller, Zuozhi Yang, Colin Messinger, Calin Anton, Karo Castro-Wunsch, William Maga, Steven Bogaerts, Robert Arrington, and Clay Langley. Parameterized Poker Squares (PPS) is a generalization of Poker Squares where players must adapt to a point system supplied at play time and thus dynamically compute highly varied strategies. The authors detail the top three performing AI players in a PPS research competition, all three of which make use of a variety of Monte Carlo techniques.

"Monte Carlo Tree Search with Robust Exploration" is authored by Takahisa Imagawa and Tomoyuki Kaneko. This paper presents a new MCTS method that focuses on identifying the best move. By minimizing the cumulative regret, UCT has achieved remarkable success in Go and other games. However, recent studies on straight-forward regret reveal that there are better exploration strategies. To improve the current performance, a leaf to be explored is determined not only by the mean but also by the whole reward distribution. The authors adopt a hybrid approach to obtain reliable distributions. A negamax-style backup of reward distributions is used in the shallower half of a search tree, and UCT is adopted in the rest of the tree. Experiments on synthetic trees show that the presented method outperformed UCT and other similar methods, except for trees having uniform width and depth.

“Pruning Playouts in Monte Carlo Tree Search for the Game of Havannah” is written by Joris Duguépéroux, Ahmad Mazyad, Fabien Teytaud, and Julien Dehos. MCTS is a popular technique for playing multi-player games. In the paper, the authors propose a new method to bias the playout policy of MCTS. The idea is to prune the decisions that seem “bad” (according to the previous iterations of the algorithm) before computing each playout. Thus, the method evaluates the estimated “good” moves more precisely. The improvement is tested for the game of Havannah and compared with several classic improvements. The method outperforms the classic version of MCTS (with the RAVE improvement) and the different playout policies of MCTS that have been submitted to experiments.

“Fast Seed-Learning Algorithms for Games” is authored by Jialin Liu, Olivier Teytaud, and Tristan Cazenave. Recently, a methodology was presented for boosting the computational intelligence of randomized game-playing programs. The authors propose faster variants of these algorithms, namely, rectangular algorithms (fully parallel) and bandit algorithms (faster in a sequential setup). They check the performance on several board games and card games. In addition, in the case of Go, they check the methodology when the opponent is completely distinct to the opponent used in the training.

“Heuristic Function Evaluation Framework” is written by Nera Nešić and Stephan Schiffel. The authors present a heuristic function evaluation framework that allows one to quickly compare a heuristic function’s output with benchmark values that are precomputed for a subset of positions in the state space of the game. The framework reduces the time to evaluate a heuristic function drastically while also providing some insight into where the heuristic is performing well or below par. The authors analyze the feasibility of using MCTS to compute benchmark values instead of relying on game theoretic values that are hard to obtain in many cases. They also propose several metrics for comparing heuristic evaluations with benchmark values and discuss the feasibility of using MCTS benchmarks with those metrics.

Concrete Games

Seven papers discussed six concrete games. They are: 2048, Werewolf Game (two articles), Mastermind, Domineering, Reverse Hex, and Computer-Aided Go.

“Systematic Selection of N-tuple Networks for 2048” is authored by Kazuto Oka and Kiminori Matsuzaki. The puzzle game 2048 is a single-player stochastic game played on a 4×4 grid. It is the most popular game among similar slide-and-merge games. One of the strongest computer players for 2048 uses temporal difference learning (TD learning) with N-tuple networks. Here, it matters a great deal how to design the N-tuple networks. In the paper, the authors thoroughly study the N-tuple networks for the game 2048. In the first set of experiments, they conduct TD learning by selecting 6- and 7-tuples exhaustively, and evaluate the usefulness of those tuples. In the second set of experiments, they conduct TD learning with high-utility tuples, varying the number of tuples. The best player with ten 7-tuples achieves good results. It utilizes no game-tree search and plays a move in about 12 microseconds.

"Human-Side Strategies in the Werewolf Game Against the Stealth Werewolf Strategy" is written by Xiaoheng Bi and Tetsuro Tanaka. The werewolf game contains unique features, such as persuasion and deception, which are not included in games that have been previously studied in AI research. The authors concentrate on a werewolf-side strategy called "stealth werewolf." With this strategy, each of the werewolf-side players behaves like a villager, and the player does not pretend to have a special role. Even though the strategy is thought to be suboptimal, this has not been proved. The authors restrict the human-side strategies such that (1) the seer reveals his/her role on the first day, (2) the bodyguard never reveals his/her role, and (3) the advantage of the werewolves in determining the player to be eliminated by vote is nullified. They calculate the ε-Nash equilibrium of strategies for both sides under these three restrictions, and discuss implications.

"Werewolf Game Modeling Using Action Probabilities Based on Play Log Analysis" is authored by Yuya Hirata, Michimasa Inaba, Kenichi Takahashi, Fujio Toriumi, Hirotaka Osawa, Daisuke Katagami, and Kousuke Shinoda. In the study, the authors construct a non-human agent that can play the werewolf game (i.e., AI wolf) with the aims of creating more advanced intelligence and acquiríng more advanced communication skills for AI-based systems. They build a behavioral model using information regarding human players and the decisions made by such players; all such information is obtained from play logs of the werewolf game. To confirm the model, simulation experiments are conducted of the werewolf game using an agent based on the proposed behavioral model, as well as a random agent for comparison. An 81.55% coincidence ratio of agent behavior versus human behavior is obtained.

"Nash Equilibrium in Mastermind" is written by François Bonnet and Simon Viennot. Mastermind is a famous two-player deduction game. A Codemaker chooses a secret code and a Codebreaker tries to guess this secret code in as few guesses as possible, with feedback information after each guess. Many existing works have computed optimal worst-case and average-case strategies of the Codebreaker, assuming that the Codemaker chooses the secret code uniformly at random. However, the Codemaker can freely choose any distribution probability on the secret codes. An optimal strategy in this more general setting is known as a Nash Equilibrium. In the current research, the authors compute such a Nash Equilibrium for all instances of Mastermind up to the most classic instance of four pegs and six colors, showing that the uniform distribution is not always the best choice for the Codemaker. They also show the direct relation between Nash Equilibrium computations and computations of worst-case and average-case strategies.

"11×11 Domineering Is Solved: The First Player Wins" is authored by Jos Uiterwijk. The author has developed a program called MUDoS (Maastricht University Domineering Solver) that solves Domineering positions in a very efficient way. MUDoS enables the solution of currently known positions (up to the 10×10 board) much quicker (measured in number of investigated nodes) than has happened to date. More importantly, MUDoS enables the solution of the 11×11 Domineering board. This board was until now far out of reach of previous Domineering solvers. The solution needed the investigation of 259,689,994,008 nodes, using almost half a year of computation time on a single simple desktop computer. The results show that under optimal play the first player wins, irrespective of whether Vertical or Horizontal starts the game. In addition, several other boards hitherto unsolved are also solved.

"A Reverse Hex Solver" is written by Kenny Young and Ryan Hayward. The authors present Solrex, an automated solver for the game of Reverse Hex. Reverse Hex, also known as Rex or Misère Hex, is the variant of the game of Hex in which the player who joins his/her two sides loses the game. Solrex performs a mini-max search of the state space using Scalable Parallel Depth First Proof Number Search, enhanced by the pruning of inferior moves and the early detection of certain winning strategies. Solrex is implemented on the same code base as the Hex program Solver, and can solve arbitrary positions on board sizes up to 6×6, with the hardest position taking less than four hours on four threads.

"Computer-Aided Go: Chess as a Role Model" is authored by Ingo Althöfer. Recently, computers have gained strength in the Asian board game Go. Similar to the experience in Chess some 15–30 years ago, teams with humans and computers may be much stronger than each of their Go components. The paper claims that time is ripe for computer-aided Go on a large scale, although so far neither most users nor the Go programmers have thought about it. A main part of the paper describes successful pioneers in playing Go with computer help. Progress in computer-aided Go may also lead to progress in human Go and in computer Go itself.

Theory and Complexity

The five topics discussed are: Polyhedral Uncertainty Set (two-person zero-sum games), A Class Grammar (General Games), the Number of Legal Go Positions (Go), A Googleplex of Games (Go), and Majority Systems (Subtraction Game).

"Quantified Integer Programs with Polyhedral Uncertainty Set" is written by Michael Hartisch, Thorsten Ederer, Ulf Lorenz, and Jan Wolf. Quantified Integer Programs (QIPs) are integer programs with variables being either existentially or universally quantified. They can be interpreted as a two-person zero-sum game with an existential and a universal player, where the existential player tries to meet all constraints and the universal player intends to force at least one constraint to be not satisfied. Originally, the universal player is only restricted to set the universal variables within their upper and lower bounds. This idea is extended by adding constraints for the universal variables, i.e., restricting the universal player to some polytope instead of the hypercube created by bounds. It is also shown how this extended structure can be reduced from a polynomial-time algorithm to a QIP.

"A Class Grammar for General Games" is authored by Cameron Browne. While there exist a variety of game description languages (GDLs) for modeling various classes of games, the GDLs discussed are aimed at game playing rather than the more particular needs of game design. The paper describes a new approach to general game modeling that arose from this need. A class grammar is automatically generated from a given library of source code, i.e., from the constructors and associated parameters found along its class hierarchy, to give a context-free grammar that provides access to the underlying code while hiding its implementation details.

"The Number of Legal Go Positions" is written by John Tromp. The number of legal 19×19 Go positions has been determined as

```
2081681993819799846994786333448627702865224
5388453054842563945682092741961273801537852
5648451698519643907259916015628128546089888
3144271297153193175577366203972470648409 35
```

which is approximately $2 \cdot 10^{170}$. This is roughly 1.2% of the total number of positions, being $3^{19\times19}$. The proof uses a correspondence between legal positions and paths through a graph of so-called border states. It requires considerable computing power, taking over 250,000 CPU-hours and 30 PB of disk IO.

"A Googolplex of Go Games" is authored by Matthieu Walraet and John Tromp. The authors establish the existence of $10^{10^{100}}$ Go games on the 19×19 board. Players can produce very long games: They fill in their eyes and continue capturing each other, restricted only by the superko rule that forbids repeating the whole board position. The challenge in proving a lower bound is to make a single game as long as possible, by visiting as many of the roughly $2 \cdot 10^{170}$ legal positions as possible. It will then turn out that there are sufficient choices along the way to lift the game length into the exponent.

"An Analysis of Majority Systems with Dependent Agents in a Simple Subtraction Game" is written by Raphael Thiele and Ingo Althöfer. It is common knowledge that a majority system is typically better than its components, when the components are stochastically independent. However, in practice the independency assumption is often not justified. The authors investigate systems of experts that are constituted by couples of dependent agents. Based on recent theoretical work, they analyze their performance in a simple two-player subtraction game. It turns out that systems with negatively correlated couples perform better than those with a positive correlation within the couples. From computer chess practice, it was known that systems of very positively correlated bots were not so successful.

Cognition Model

One paper is classified under the heading Cognition Model. Still, it is an important topic that clearly belongs to this conference.

"Do People Think Like Computers?" is authored by Bas van Opheusden, Zahy Bnaya, Gianni Galbiati, and Wei Ji Ma. From computer-chess practice it is known that systems of rather positively correlated bots are not always successful, since they run the risk of missing an important variation. At first, human cognition inspired the earliest algorithms for game-playing computer programs. Then, however, the studies of human and computer game play quickly diverged: the artificial intelligence (AI) community focused on theory and techniques to solve games, while behavioral scientists empirically examined the specific topic of simple decision-making in humans. In this paper, the authors combine concepts and methods from the two fields to investigate whether human and AI players take similar approaches in an adversarial combinatorial game. The authors develop and compare several models that capture human behavior, and demonstrate that the models can predict behavior in two related tasks. At the end, they use the models to describe what makes a strong human player.

The book would not have been produced without the help of many persons. In particular, we would like to mention the authors and the reviewers for their help. Moreover, the organizers of the three events in Leiden (see the beginning of this preface) have contributed substantially by bringing the researchers together. Without much emphasis, we recognize the work by the various committees of CG 2016 as essential for this publication. One exception is made for Joke Hellemons, who is gratefully thanked for all services to our games community. Finally, the editors happily recognize the generous sponsors: NWO Exact Sciences, Museum Naturalis, Surf-SARA, Municipality of Leiden, Digital Games Technology, Faculty of Science, ICGA, ISSC, the Leiden Institute of Advanced Computer Science, and the Leiden Centre of Data Science.

September 2016

Aske Plaat
Jaap van den Herik
Walter Kosters

http://www.springer.com/978-3-319-50934-1

Computers and Games
9th International Conference, CG 2016, Leiden, The Netherlands, June 29 – July 1, 2016, Revised Selected Papers
Plaat, A.; Kosters, W.; van den Herik, J. (Eds.)
2016, XX, 225 p. 68 illus., Softcover
ISBN: 978-3-319-50934-1