

Preface

This book contains the papers of the 8th Computers and Games Conference (CG 2013) held in Yokohama, Japan. The conference took place during August 13 to August 15, 2013 in conjunction with the 17th Computer and Games Tournament and the 20th 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. The Yokohama conference was definitively characterized by the progress of Monte Carlo Tree Search (MCTS) and the development of new games. Earlier conferences took place in Tsukuba (1998), Hamamatsu (2000), Edmonton (2002), Ramat-Gan (2004), Turin (2006), Beijing (2008), and Kanazawa (2010).

The Program Committee (PC) was pleased to see that so much progress was made in MCTS and that on top of that new games and new techniques were added to the recorded achievements. Each paper was sent to at least three referees. If conflicting views on a paper were reported, the referees themselves arrived at a proper decision. With the help of many referees (see after the preface), the PC accepted 21 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 (i) to give authors the opportunity to include the results of the fruitful discussion after the lecture into their paper, and (ii) to maintain the high quality threshold of the ACG series. The authors enjoyed this procedure.

The above-mentioned set of 21 papers covers a wide range of computer games and many different research topics. We have grouped the topics into five classes. We mention the classes in the order of publication (with the exception of one paper): Monte Carlo Tree Search and its enhancements (7 papers), solving and searching (7 papers), analysis of a game characteristic (5 papers), new approaches (1 paper), and serious games (1 paper).

We hope that the readers will enjoy the research efforts performed by the authors. Below we reproduce brief characterizations of the 21 contributions taken from the text as submitted by the authors. The authors of the first two publications “Dependency-Based Search for Connect6” and “On Semeai Detection in Monte-Carlo Go” received the shared Best Paper Award of the CG2013. As a courtesy to the award winners we start with a paper on Solving and Searching.

“Dependency-based Search for Connect6” is a contribution by I-Chen Wu, Hao-Hua Kang, Hung-Hsuan Lin, Ping-Hung Lin, Ting-Han Wei, Chieh-Min Chang, and Ting-Fu Liao. In 1994, Victor Allis *et al.* proposed dependency-based search (DBS) to solve Go-Moku, a kind of five-in-a-row game. DBS is critical for threat space search (TSS) when there are many independent or nearly independent TSS areas. Similarly, DBS is also important for the game Connect6, a kind of six-in-a-row game with two pieces per move. However, the rule that two pieces are played per move in Connect6

makes DBS rather difficult to apply in Connect6 programs. This paper is the first attempt to apply DBS in Connect6 programs. The targeted program is NCTU6, which won Connect6 tournaments in the Computer Olympiad twice and defeated many professional players in Man-Machine Connect6 championships. The experimental results show that DBS yields a speedup factor of 4.12 on average, and up to 50 for some hard positions.

“On Semeai Detection in Monte-Carlo Go” by Tobias Graf, Lars Schaefer, and Marco Platzner describes the inability of Monte-Carlo Tree Search (MCTS) based Go programs to recognize and adequately handle capturing races, also known as semeai, especially when many of them appear simultaneously. The inability essentially stems from the fact that certain semeai require deep lines of correct tactical play which is not directly related to the exploratory nature of MCTS. In this paper the authors provide a technique for heuristically detecting and analyzing semeai during the search process of a state-of-the-art MCTS implementation. The strength of the approach is evaluated on game positions that are known to be difficult to handle even by the strongest Go programs to date. The results show a clear identification of semeai and thereby advocate the approach as a promising heuristic for the design of future MCTS simulation policies.

“Efficiency of Static Knowledge Bias in Monte-Carlo Tree Search” is a contribution by Kokoro Ikeda and Simon Viennot. Currently, Monte-Carlo methods are the best known algorithms for the game of Go. The Monte-Carlo simulations based on a probability model containing static knowledge of the game have been shown to be more efficient than random simulations. Some programs also use such probability models in the tree search policy to limit the search to a subset of the legal moves or to bias the search. However, this aspect is not so well documented. In the paper, the authors describe more precisely how static knowledge can be used to improve the tree search policy. They experimentally show the efficiency of the proposed method by a large number of games played against open source Go programs.

“Investigating the Limits of Monte-Carlo Tree Search Methods in Computer Go” is authored by Shih-Chieh Huang and Martin Müller. Obviously, Monte-Carlo Tree Search methods have led to huge progress in computer Go. Still, program performance is uneven - most current Go programs are much stronger in some aspects of the game, such as local fighting and positional evaluation, than in other aspects. Well known weaknesses of many programs include (1) the handling of several simultaneous fights, including the ‘two safe groups’ problem, and (2) the dealing with coexistence in seki. After a brief review of MCTS techniques, three research questions regarding the behavior of MCTS-based Go programs in specific types of Go situations are formulated. Then, an extensive empirical study of 10 leading Go programs investigates their performance of two specifically designed test sets containing ‘two safe groups’ and seki situations. The results give a good indication of the state-of-the-art in computer Go as of 2012/2013. They show that while a few of the very top programs can apparently solve most of these evaluation problems in their playouts already, these problems are difficult to solve by global search.

“Programing Breakthrough” by Richard Lorentz and Therese Horey analyzes the abstract strategy board game Breakthrough. The game requires a well balanced attention for strategical issues in the early stages but can suddenly and unexpectedly

turn tactical. The strategic elements can be extremely subtle and the tactics can be quite deep, involving sequences of 20 or more moves. The authors thoroughly analyze new and existing features of an MCTS-based program to play Breakthrough. They demonstrate that this approach, with proper adjustments, is quite successful. The current version can beat most human players.

“MoHex 2.0: a Pattern-based MCTS Hex Player”, is a contribution by Shih-Chieh Huang, Broderick Arneson, Ryan Hayward, Martin Müller, and Jakub Pawlewicz. It deals with the Monte-Carlo tree search revolution which has spread from computer Go to many areas, including computer Hex. MCTS Hex players are now on par with traditional knowledge-based alpha-beta search players. In this paper the reigning Computer Olympiad Hex gold medallist MoHex, an MCTS player, is strengthened by several improvements leading to MoHex2. The first improvement is replacing a hand-crafted MCTS simulation policy by one based on learned local patterns. Two other improvements are the applications of the minorization-maximization algorithm. The resulting pattern weights are used in both the leaf selection and the simulation phases of MCTS. All these improvements can increase the playing strength considerably, since the resultant MoHex2.0 is about 250 Elo points stronger than MoHex.

“Analyzing Simulations in Monte-Carlo Tree Search for the Game of Go” is a contribution by Sumudu Fernando and Martin Müller. In Monte-Carlo Tree Search, simulations (or playouts) play a crucial role since they replace the evaluation function used in classical game-tree search and guide the development of the game tree. Despite their importance, not too much is known about the details of how they work. This paper starts a more in-depth study of simulations, using the game of Go, and in particular the program Fuego, as an example. Playout policies are investigated in terms of the number of blunders they make, and in terms of how many points they lose over the course of a simulation. The result is a deeper understanding of the different components of the Fuego playout policy. Consequently, 7 suggestions for closer examination are provided. Finally, a list of fundamental questions about simulation policies is given.

“Anomalies of Pure Monte-Carlo Search in Monte Carlo Perfect Games” written by Ingo Althöfer and Wesley Michael Turner is an interesting paper that ends with 6 open problems. So, the reader is encouraged to read this inspiring research paper and try to find an own contribution. A game is called “Monte-Carlo perfect” when in each position pure Monte-Carlo search converges to perfect play as the number of simulations tend toward infinity. The authors exhibit three families of Monte-Carlo perfect single-player and two-player games where this convergence is not monotonic. Moreover, they give a family of MC-perfect games in which MC(1) performs arbitrarily well against MC(1000).

“Developments on Product Propagation” is written by Abdallah Saffidine and Tristan Cazenave. Product Propagation (PP) is an algorithm to backup probabilistic evaluations for abstract two-player games. So far, it was shown that PP could solve Go problems as efficiently as Proof Number Search (PNS). In this paper, the authors exhibit three domains where PP performs better (see the nuances in the paper) than previously known algorithms for solving games. The compared approaches include alpha-beta search, PNS, and Monte Carlo Tree Search. The authors also extend PP to deal with its memory consumption and to improve its solving time.

“Solution Techniques for Quantified Linear Programs and the Links to Gaming” is a contribution by Ulf Lorenz, Thomas Opfer, and Jan Wolf. Quantified linear programs (QLPs) are linear programs (LPs) with variables being either existentially or universally quantified. QLPs are two-person zero-sum games between an existential and a universal player on the one side, and convex multistage decision problems on the other side. Solutions of feasible QLPs are so called winning strategies for the existential player that specify how to react on moves – well-thought fixations of universally quantified variables – of the universal player to be sure to win the game. To find a certain best strategy among different winning strategies, we propose the extension of the QLP decision problem by an objective function. To solve the resulting QLP optimization problem, we exploit the problem’s hybrid nature and combine linear programming techniques with solution techniques from game-tree search. As a result, we present an extension of the Nested Benders Decomposition algorithm by the $\alpha\beta$ -algorithm and its heuristical move-ordering as used in game-tree search to solve minimax trees. The applicability of our method to both QLPs and models of PSPACE-complete games such as Connect6 is examined in an experimental evaluation.

“Improving Best-reply Search” is a contribution by Markus Esser, Michael Gras, Mark Winands, Maarten Schadd, and Marc Lanctot. Best-Reply Search (BRS) is a new search technique for game-tree search in multi-player games. In BRS, the exponentially many possibilities that can be considered by opponent players is flattened so that only a single move, the best one among all opponents, is chosen. BRS has been shown to outperform the classic search techniques in several domains. However, BRS may consider invalid game states. In this paper, the authors improve the BRS search technique such that it preserves the proper turn order during the search and does not lead to invalid states. The new technique, BRS+, uses the move ordering to select moves at opponent nodes that are not searched. Empirically, they show that BRS+ significantly improves the performance of BRS in Multi-Player Chess, leading to winning 8.3 percent to 11.1 percent more games against the classic techniques maxn and Paranoid, respectively. When BRS+ plays against maxn, Paranoid, and BRS at once, it wins the most games as well.

“Scalable Parallel DFPN Search”, written by Jakub Pawlewicz and Ryan Hayward discusses a new shared-memory parallel version of depth-first proof number search. Based on the serial DFPN $1 + \epsilon$ method of Pawlewicz and Lew, SPDFPN searches effectively even as the transposition table becomes almost full, and so can solve large problems. To assign jobs to threads, SPDFPN uses proof and dis-proof numbers. It uses no domain-specific knowledge or heuristics, so it can be used in any domain. The authors tested SPDFPN on problems from the game of Hex. SPDFPN performs well on our scalability test: on a 24-core machine with a task taking 4.2 hours on a single thread, parallel efficiency ranges from about 0.8 on 4 threads to about 0.74 on 16 threads. SPDFPN also performs well on harder problems: it solved all previously intractable 9×9 Hex opening moves, with the hardest opening taking about 111 days. Also, in 63 days, it solved one 10×10 Hex opening move. It is the first time that a computer (or a human) has solved a 10×10 Hex opening move. The current state of the art in Hex solving may fascinate experts and laymen alike.

“A Quantitative Study of 2×4 Chinese Dark Chess” is authored by Hung-Jui Chang and Tsan-sheng Hsu. In this paper, the authors study Chinese dark chess (CDC), a popular 2-player imperfect information game, that is a variation of Chinese chess played on a 2×4 gameboard. The 2×4 version is solved by computing the exact value of each board position for all possible fair piece combinations. The results of the experiments demonstrate that the initial arrangement of the pieces and the place to reveal the first piece are the most important factors to affect the outcome of a game.

“Cylinder-Infinite-Connect-Four except for Widths 2, 6, and 11 is Solved: Draw” is written by Yoshiaki Yamaguchi, Tetsuro Tanaka, and Kazunori Yamaguchi. Cylinder-Infinite-Connect-Four is a variant of the Connect-Four game played on cylindrical boards of differing cyclic widths and infinite height. In this paper, the authors show strategies to avoid losing at Cylinder-Infinite-Connect-Four except for Widths 2, 6, and 11. If both players use the strategies, the game will be drawn. This result can also be used to show that Width-Limited-Infinite-Connect-Four is drawn for any width. Finally, the authors show that Connect-Four of any size with passes allowed is drawn.

“Havannah and TwixT are PSPACE-complete” is authored by Édouard Bonnet, Florian Jamain, and Abdallah Saffidine. Numerous popular abstract strategy games ranging from Hex and Havannah to Lines of Action belong to the class of connection games. Still, very few complexity results on such games have been obtained since Hex was proved pspace-complete in the early 1980s. The authors study the complexity of two connection games among the most widely played ones, i.e., they prove that Havannah and TwixT are pspace-complete. The proof for Havannah involves a reduction from Generalized Geography and is based solely on ring-threats to represent the input graph. The reduction for TwixT builds upon previous work as it is a straightforward encoding of Hex.

“Material Symmetry to Partition Endgame Tables”, by Abdallah Saffidine, Nicolas Jouandeau, Cédric Buron, and Tristan Cazenave describes that many games display some kind of material symmetry. That is, some sets of game elements can be exchanged for another set of game elements, so that the resulting position will be equivalent to the original one, no matter how the elements were arranged on the board. Material symmetry is routinely used in card game engines when they normalize their internal representation of the cards. Other games such as Chinese dark chess also feature some form of material symmetry, but it is much less clear what the normal form of a position should be. The authors propose a principled approach to detect material symmetry. Their approach is generic and is based on solving multiple relatively small subgraph isomorphism problems. They show how it can be applied to Chinese dark chess, dominoes, and skat. In the latter case, the mappings obtained are equivalent to the ones resulting from the standard normalization process. In the two former cases, the authors show that the material symmetry allows for impressive savings in memory requirements when building endgame tables. They also show that those savings are relatively independent of the representation of the tables.

“Further Investigations of 3-Member Simple Majority Voting for Chess” is a contribution by Kristian Toby Spoerer, Toshihisa Okaneya, Kokolo Ikeda, and Hiroyuki Iida. The 3-member simple majority voting is investigated for the game of Chess. The programs Stockfish, TogaII and Bobcat are used. Games are played against

the strongest member of the group and against the group using simple majority voting. The authors show that the group is stronger than the strongest program. Subsequently, they investigate the following research question: “under what conditions is 3-member simple majority voting stronger than the strongest member?” To answer this question the authors performed experiments on 27 groups. Statistics were gathered on the situations where the group outvoted the group leader. Two conditions were found. First, group members should be almost equal in strength, while still showing a small, but significant strength difference. Second, the denial percentage of the leader’s candidate move depends on the strength of the members.

“Comparison Training of Shogi Evaluation Functions with Self-Generated Training Positions and Moves” is written by Akira Ura, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. Automated tuning of parameters in computer game playing is an important technique for building strong computer programs. Comparison training is a supervised learning method for tuning the parameters of an evaluation function. It has proven to be effective in the game of Chess and Shogi. The training method requires a large number of training positions and moves extracted from game records of human experts; however, the number of such game records is limited. In this paper, they propose a practical approach to creating additional training data for comparison training by using the program itself. They investigate three methods for generating additional positions and moves. Then they evaluate them using a Shogi program. Experimental results show that the self-generated training data can improve the playing strength of the program.

“Automatic Generation of Opening Books for Dark Chess” by Bo-Nian Chen and Tsan-sheng Hsu describes that playing the opening game of Chinese dark chess well is a challenge that depends highly on probability. So far there are no known studies or published results for opening game studies. Although automatic open book generation is a common research topic in many games. Some researchers collect master games to build an opening book; others automatically gather computer-played games as their open books. However, in Chinese dark chess, it is still hard to obtain a strong opening book by the above strategies because few master games have been recorded. In this paper, the authors propose a policy-oriented search to build automatically a selective open book that is helpful in practical game playing. The constructed open book provides positive feedback for computer programs that play Chinese dark chess.

“Optimal, Approximately Optimal, and Fair Play of the Fowl Play Card Game” by Todd Neller, Marcin Malec, Clifton Presser, and Forrest Jacobs analyzes optimal play of the jeopardy card game Fowl Play. The paper starts by presenting play that maximizes the expected score per turn, explaining why this differs from optimal play. After describing the equations of optimal play and the techniques used to solve them, the authors present visualizations of such play and compare them to optimal play of a simpler, related dice game Pig. Next, they turn our attention to the use of function approximation in order to demonstrate the feasibility of a good, memory-efficient approximation of optimal play. Finally, they apply the analytical techniques towards the parameterization, tuning, and improved redesign of the game with komi for optimal fairness.

“Resource, Entity, Action: A Generalized Design Pattern for RTS Games” is authored by Mohamed Abbadi, Francesco Di Giacomo, Renzo Orsini, Aske Plaat,

Pieter Spronck, and Giuseppe Maggiore. In many Real-Time Strategy (RTS) games, players develop an army in real time, then attempt to take out one or more opponents. Despite the existence of basic similarities among the many different RTS games, engines of these games are often built ad hoc, and code re-use among different titles is minimal. The authors created a design pattern called “Resource Entity Action” (REA) abstracting the basic interactions that entities have with each other in most RTS games. The paper discusses the REA pattern and its language abstraction. The authors also discuss the implementation in the Casanova game programming language. Their analysis shows that the pattern forms a solid basis for a playable RTS game, and that it achieves considerable gains in terms of lines of code and runtime efficiency. The conclusion is that the REA pattern is a suitable approach to the implementation of many RTS games.

Acknowledgments

This book would not have been produced without the help of many people. In particular, we would like to mention the authors and the referees for their help. Moreover, the organizers of the three events in Yokohama (see the beginning of this preface) contributed substantially by bringing the researchers together. Without much emphasis, we recognize the work by the committees of the CG 2013 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 The Brain and Mind-Sports Foundation, JAIST, Tilburg University, Tilburg center for Cognition and Communication, ICGA, and Digital Games Technology.

April 2014

H. Jaap van den Herik
Hiroyuki Iida
Aske Plaat

Computers and Games

8th International Conference, CG 2013, Yokohama,
Japan, August 13-15, 2013, Revised Selected Papers
van den Herik, H.J.; Iida, H.; Plaat, A. (Eds.)

2014, XX, 257 p. 131 illus., Softcover

ISBN: 978-3-319-09164-8