Preface

This book contains the papers of the 14th Advances in Computer Games Conference (ACG 2015) held in Leiden, The Netherlands. The conference took place during July 1–3, 2015, in conjunction with the 18th Computer Olympiad and the 21st World Computer-Chess Championship.

The Advances in Computer Games conference series is a major international forum for researchers and developers interested in all aspects of artificial intelligence and computer game playing. As with many of the recent conferences, this conference saw a large number of papers with progress in Monte Carlo tree search. However, the Leiden conference also showed diversity in the topics of papers, e.g., by including papers on thorough investigations on search, theory, complexity, games (performance and prefiling), and machine learning. Earlier conferences took place in London (1975), Edinburgh (1978), London (1981, 1984), Noordwijkerhout (1987), London (1990), Maastricht (1993, 1996), Paderborn (1999), Graz (2003), Taipei (2005), Pamplona (2009), and Tilburg (2011).

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, 36 authors submitted a paper. 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 22 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 threshold of the ACG series. The authors enjoyed this procedure.

The aforementioned set of 22 papers covers a wide range of computer games and many different research topics. We grouped the topics into the following five classes according to the order of publication: Monte Carlo tree search (MCTS) and its enhancements (six papers), theoretical aspects and complexity (five papers), analysis of game characteristics (five papers), search algorithms (three papers), and machine learning (three papers).

We hope that the readers will enjoy the research efforts presented by the authors. Here, we reproduce brief characterizations of the 22 contributions largely relying on the text as submitted by the authors. The idea is to show a connection between the contributions and insights into the research progress.

Monte Carlo Tree Search

The authors of the first publication "Adaptive Playouts in Monte Carlo Tree Search with Policy Gradient Reinforcement Learning" received the ACG 2015 Best Paper Award. The paper is written by Tobias Graf and Marco Platzner. As the title suggests, the paper introduces an improvement in the playout phase of MCTS. If the playout policy evaluates a position wrongly then cases may occur where the tree search faces severe difficulties to find the correct move; a large search space may be entered unintentionally. The paper explores adaptive playout policies. The intention is to improve the playout policy *during* the tree search. With the help of policy-gradient reinforcement learning techniques the program optimizes the playout policy to arrive at better evaluations. The authors provide tests for computer Go and show an increase in playing strength of more than 100 ELO points.

"Early Playout Termination in MCTS," authored by Richard Lorentz, also deals with the playout phase of MCTS. Nowadays minimax-based and MCTS-based searches are seen as competitive and incompatible approaches. For example, it is generally agreed that chess and checkers require a minimax approach while Go and Havannah are better off by MCTS. However, a hybrid technique is also possible. It works by stopping the random MCTS playouts early and using an evaluation function to determine the winner of the playout. This algorithm is called MCTS-EPT (MCTS with early playout termination), and is studied in this paper by using MCTS-EPT programs written for Amazons, Havannah, and Breakthrough.

"Playout Policy Adaptation for Games," a contribution by Tristan Cazenave, is the third paper on playouts in MCTS. For general game playing, MCTS is the algorithm of choice. The paper proposes to learn a playout policy online in order to improve MCTS for general game playing. The resulting algorithm is named Playout Policy Adaptation, and was tested on Atarigo, Breakthrough, Misere Breakthrough, Domineering, Misere Domineering, Go, Knightthrough, Misere Knightthrough, Nogo, and Misere Nogo. For most of these games, Playout Policy Adaptation is better than UCT for MCTS with a uniform random playout policy, with the notable exceptions of Go and Nogo.

"Strength Improvement and Analysis for an MCTS-Based Chinese Dark Chess Program" is authored by Chu-Hsuan Hsueh, I-Chen Wu, Wen-Jie Tseng, Shi-Jim Yen, and Jr-Chang Chen. The paper describes MCTS and its application in the game of Chinese Dark Chess (CDC), with emphasis on playouts. The authors study how to improve and analyze the playing strength of an MCTS-based CDC program, named DarkKnight, which won the CDC tournament in the 17th Computer Olympiad. They incorporated three recent techniques, early playout terminations, implicit minimax backups, and quality-based rewards. For early playout terminations, playouts end when reaching states with likely outcomes. Implicit minimax backups use heuristic evaluations to help guide selections of MCTS. Quality-based rewards adjust rewards based on online collected information. Experiments show that the win rates against the original DarkKnight are 60 %, 70 %, and 59 %, respectively, when incorporating the three techniques. By incorporating all three together, the authors obtain a win rate of 77 %.

"LinUCB Applied to Monte Carlo Tree Search" is written by Yusaku Mandai and Tomoyuki Kaneko. This paper deals with the selection function of MCTS; as is well known, UCT is a standard selection method for MCTS algorithms. The study proposes a family of LinUCT algorithms (linear UCT algorithm) that incorporate LinUCB into MCTS algorithms. LinUCB is a recently developed method. It generalizes past episodes by ridge regression with feature vectors and rewards. LinUCB outperforms UCB1 in contextual multi-armed bandit problems. The authors introduce a straightforward application of LinUCB, called $\text{LinUCT}_{\text{PLAIN}}$. They show that it does not work well owing to the minimax structure of game trees. Subsequently, they present $\text{LinUCT}_{\text{RAVE}}$ and $\text{LinUCT}_{\text{FP}}$ by further incorporating two existing techniques, rapid action value estimation (RAVE) and feature propagation (FP). Experiments were conducted by a synthetic model. The experimental results indicate that $\text{LinUCT}_{\text{RAVE}}$, $\text{LinUCT}_{\text{FP}}$, and their combination $\text{LinUCT}_{\text{RAVE-FP}}$ all outperform UCT, especially when the branching factor is relatively large.

"Adapting Improved Upper Confidence Bounds for Monte-Carlo Tree Search" is authored by Yun-Ching Liu and Yoshimasa Tsuruoka. The paper asserts that the UCT algorithm, which is based on the UCB algorithm, is currently the most widely used variant of MCTS. Recently, a number of investigations into applying other bandit algorithms to MCTS have produced interesting results. In their paper, the authors investigate the possibility of combining the improved UCB algorithm, proposed by Auer et al., with MCTS. The paper describes the Mi-UCT algorithm, which applies the modified UCB algorithm to trees. The performance of Mi-UCT is demonstrated on the games of 9×9 Go and 9×9 NoGo. It was shown to outperform the plain UCT algorithm when only a small number of playouts are given, while performing roughly on the same level when more playouts are available.

Theory and Complexity

"On Some Random Walk Games with Diffusion Control" is written by Ingo Althöfer, Matthias Beckmann, and Friedrich Salzer. This paper is not directly about MCTS, but contains a theoretical study of an element related to MCTS: the effect of random movements. The paper remarks that random walks with discrete time steps and discrete state spaces have widely been studied for several decades. The authors investigate such walks as games with "diffusion control": a player (controller) with certain intentions influences the random movements of a particle. In their models the controller decides only about the step size for a single particle. It turns out that this small amount of control is sufficient to cause the particle to stay in "premium regions" of the state space with surprisingly high probabilities.

"On Some Evacuation Games with Random Walks" is a contribution by Matthias Beckmann. The paper contains a theoretical study of a game with randomness. A single player game is considered; a particle on a board has to be steered toward evacuation cells. The actor has no direct control over this particle but may indirectly influence the movement of the particle by blockades. Optimal blocking strategies and the recurrence property are examined experimentally. It is concluded that the random walk of the game is recurrent. Furthermore, the author describes the average time in which an evacuation cell is reached.

"Go Complexities" is authored by Abdallah Saffidine, Olivier Teytaud, and Shi-Jim Yen. The paper presents a theoretical study about the complexity of Go. The game of Go is often said to be EXPTIME-complete. This result refers to classic Go under Japanese rules, but many variants of the problem exist and affect the complexity. The authors survey what is known about the computational complexity of Go and highlight challenging open problems. They also propose new results. In particular, the authors show that Atari-Go is PSPACE-complete and that hardness results for classic Go carry over to their partially observable variant.

"Crystallization of Domineering Snowflakes" is written by Jos W.H.M. Uiterwijk. The paper contains a theoretical analysis of the game of Domineering. The author presents a combinatorial game-theoretic analysis of special Domineering positions. He investigates complex positions that are aggregates of simpler components and linked via bridging squares. Two theorems are introduced. They state that an aggregate of two components has as its game-theoretic value the sum of the values of the components. These theorems are then extended to deal with the case of multiple-connected networks of components. As an application, an interesting Domineering position is introduced, consisting of a $*2$ subgame (star-two subgame) with four $*$ components attached: the so-called Snowflake position. The author then shows how from this Snowflake, larger chains of Snowflakes can be built with known values, including flat networks of Snowflakes (a kind of crystallization).

"First Player's Cannot-Lose Strategies for Cylinder-Infinite-Connect-Four with Widths 2 and 6" is a contribution by Yoshiaki Yamaguchi and Todd Neller. The paper introduces a theoretical result on a variant of Connect-Four. Cylinder-Infinite-Connect-Four is Connect-Four played on a cylindrical square grid board with infinite row height and columns that cycle about its width. In previous work by the same authors, the first player's cannot-lose strategies were discovered for all widths except 2 and 6, and the second player's cannot-lose strategies have been discovered for all widths except 6 and 11. In this paper, the authors show the first player's cannot-lose strategies for widths 2 and 6.

Analysis of Game Characteristics

"Development of a Program for Playing Progressive Chess" is written by Vito Janko and Matej Guid. They present the design of a computer program for playing Progressive Chess. In this game, rather than just making one move per turn, players play progressively longer series of moves. The program follows the generally recommended strategy for this game, which consists of three phases: (1) looking for possibilities to checkmate the opponent, (2) playing generally good moves when no checkmate can be found, and (3) preventing checkmates from the opponent. In their paper, the authors focus on efficiently searching for checkmates, putting to test various heuristics for guiding the search. They also present the findings of self-play experiments between different versions of the program.

"A Comparative Review of Skill Assessment: Performance, Prediction, and Profiling" is a contribution by Guy Haworth, Tamal Biswas, and Ken Regan. The paper presents results on the skill assessment of chess players. The authors argue that the assessment of chess players is both an increasingly attractive opportunity and an unfortunate necessity. Skill assessment is important for the chess community to limit potential reputational damage by inhibiting cheating and unjustified accusations of cheating. The paper argues that there has been a recent rise in both. A number of counter-intuitive discoveries have been made by benchmarking the intrinsic merit of players' moves. The intrinsic merits call for further investigation. Is Capablanca actually, objectively the most accurate world champion? Has ELO-rating inflation not taken place? Stimulated by FIDE/ACP, the authors revisit the fundamentals of the subject. They aim at a framework suitable for improved standards of computational experiments and more precise results. The research is exemplary for other games and domains. They look at chess as a demonstrator of good practice, including (1) the rating of professionals who make high-value decisions under pressure, (2) personnel evaluation by multichoice assessment, and (3) the organization of crowd-sourcing in citizen science projects. The "3P" themes of performance, prediction, and profiling pervade all these domains.

"Boundary Matching for Interactive Sprouts" is written by Cameron Browne. The paper introduces results for the game of Sprouts. The author states that the simplicity of the pen-and-paper game Sprouts hides a surprising combinatorial complexity. He then describes an optimization called boundary matching that accommodates this complexity. It allows move generation for Sprouts games of arbitrary size at interactive speeds. The representation of Sprouts positions plays an important role. "Draws, Zugzwangs, and PSPACE-Completeness in the Slither Connection Game" is written by Édouard Bonnet, Florian Jamain, and Abdallah Saffidine. The paper deals with a connection game: Slither. Two features set Slither apart from other connection games: (1) previously played stones can be displaced and (2) some stone configurations are forbidden. The standard goal is still connecting opposite edges of a board. The authors show that the interplay of the peculiar mechanics results in a game with a few properties unexpected among connection games; for instance, the existence of mutual Zugzwangs. The authors establish that (1) there are positions where one player has no legal move, and (2) there is no position where both players lack a legal move. The latter implies that the game cannot end in a draw. From the viewpoint of computational complexity, it is shown that the game is PSPACE-complete. The displacement rule can indeed be tamed so as to simulate a Hex game on a Slither board.

"Constructing Pin Endgame Databases for the Backgammon Variant Plakoto" is authored by Nikolaos Papahristou and Ioannis Refanidis. The paper deals with the game of Plakoto, a variant of backgammon that is popular in Greece. It describes the ongoing project PALAMEDES, which builds expert bots that can play backgammon variants. So far, the position evaluation relied only on self-trained neural networks. The authors report their first attempt to augment PALAMEDES with databases for certain endgame positions for the backgammon variant Plakoto. They mention five databases containing 12,480,720 records in total that can calculate accurately the best move for roughly $3.4 \cdot 10^{15}$ positions.

Search Algorithms

"Reducing the Seesaw Effect with Deep Proof-Number Search" is a contribution by Taichi Ishitobi, Aske Plaat, Hiroyuki Iida, and Jaap van den Herik. The paper studies the search algorithm Proof-Number (PN) search. In the paper, DeepPN is introduced. DeepPN is a modified version of PN search, providing a procedure to handle the seesaw effect. DeepPN employs two values associated with each node: the usual proof number and a deep value. The deep value of a node is defined as the depth to which each child node has been searched. By mixing the proof numbers and the deep value, DeepPN works with two characteristics: the best-first manner of search (equal to the original PN search) and the depth-first manner. By adjusting a parameter (called R in this paper) one can choose between best-first or depth-first behavior. In their experiments, the authors try to find a balance between both manners of searching. As it turns out, best results are obtained at an R value in between the two extremes of best-first search (original PN search) and depth-first search.

"Feature Strength and Parallelization of Sibling Conspiracy Number Search" is written by Jakub Pawlewicz and Ryan Hayward. The paper studies a variant of Conspiracy Number Search, a predecessors of Proof-Number search, with different goals. Recently the authors introduced Sibling Conspiracy Number Search (SCNS). This algorithm is not based on evaluation of the leaf nodes of the search tree but, it handles for each node, the relative evaluation scores of all children of that node. The authors implemented an SCNS Hex bot. It showed the strength of SCNS features: most critical is the initialization of the leaves via a multi-step process. The authors also investigated a simple parallel version of SCNS: it scales well for two threads but was less efficient for four or eight threads.

"Parameter-Free Tree Style Pipeline in Asynchronous Parallel Game-Tree Search" is authored by Shu Yokoyama, Tomoyuki Kaneko, and Tetsuro Tanaka. This paper describes results in parameter-free parallel search algorithms. The paper states that asynchronous parallel game-tree search methods are effective in improving playing strength by using many computers connected through relatively slow networks. In game-position parallelization, a master program manages the game tree and distributes positions in the tree to workers. Then, each worker asynchronously searches for the best move and evaluation in the assigned position. The authors present a new method for constructing an appropriate master tree that provides more important moves with more workers on their subtrees to improve playing strength. Their contribution goes along with two advantages: (1) it is parameter free in that users do not need to tune parameters through trial and error, and (2) the efficiency is suitable even for short-time matches, such as one second per move. The method was implemented in a top-level chess program (STOCKFISH). Playing strength was evaluated through self-plays. The results show that playing strength improves up to 60 workers.

Machine Learning

"Transfer Learning by Inductive Logic Programming" is a contribution by Yuichiro Sato, Hiroyuki Iida, and Jaap van den Herik. The paper studies Transfer Learning between different types of games. In the paper, the authors propose a Transfer Learning method by Inductive Logic Programming for games. The method generates general knowledge from a game, and specifies the knowledge in the form of heuristic functions, so that it is applicable in another game. This is the essence of Transfer Learning. The authors illustrate the working of Transfer Learning by taking knowledge from Tic-tac-toe and transferring it to Connect-Four and Connect-Five.

"Developing Computer Hex Using Global and Local Evaluation Based on Board Network Characteristics" is written by Kei Takada, Masaya Honjo, Hiroyuki Iizuka and Masahito Yamamoto. The paper describes a learning method for evaluation functions. The authors remark that one of the main approaches to develop a computer Hex program (in the early days) was to use an evaluation function of the electric circuit model. However, such a function evaluates the board states from one perspective only. Moreover this method has recently been defeated by MCTS approaches. In the paper, the authors propose a novel evaluation function that uses network characteristics to capture features of the board states from *two* perspectives. The proposed evaluation function separately evaluates (1) the board network and (2) the shortest path network using betweenness centrality. Then the results of these evaluations are combined. Furthermore, the proposed method involves changing the ratio between global and local evaluations through a Support Vector Machine (SVM). The new method yields an improved strategy for Hex. The resultant program, called EZO, is tested against the world-champion Hex program called MOHEX, and the results show that the method is superior to the 2011 version of MOHEX on an 11×11 board.

"Machine-Learning of Shape Names for the Game of Go" is authored by Kokolo Ikeda, Takanari Shishido, and Simon Viennot. The paper discusses the learning of shape names in Go. It starts stating that Computer Go programs with only a 4-stone handicap have recently defeated professional humans. The authors argue that by this performance the strength of Go programs is sufficiently close to that of humans, so that a new target in artificial intelligence is at stake, namely, developing programs able to provide commentary on Go games. A fundamental difficulty in achieving the new target is learning the terminology of Go, which is often not well defined. An example is the problem of naming shapes such as Atari, Attachment, or Hane. In their research, the goal is to allow a program to label relevant moves with an associated shape name. The authors use machine learning to deduce these names based on local patterns of stones. First, strong amateur players recorded for each game move the associated shape name, using a pre-selected list of 71 terms. Next, these records were used to train a supervised machine learning algorithm. The result was a program able to output the shape name from the local patterns of stones. Humans agree on a shape name with a performance rate of about 82 %. The algorithm achieved a similar performance, picking the name most preferred by humans with a rate of also about 82 %. The authors state that this is a first step toward a program that is able to communicate with human players in a game review or match.

This 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 committees of the ACG 2015 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 AEGON, NWO, the Museum Boerhaave, SurfSARA, the Municipality of Leiden, the Leiden Institute of Advanced Computer Science, the Leiden Centre of Data Science, ICGA, and Digital Games Technology.

September 2015

Aske Plaat
Jaap van den Herik
Walter Kosters

http://www.springer.com/978-3-319-27991-6

Advances in Computer Games
14th International Conference, ACG 2015, Leiden, The Netherlands, July 1-3, 2015, Revised Selected Papers
Plaat, A.; van den Herik, J.; Kosters, W. (Eds.)
2015, XX, 261 p. 128 illus. in color., Softcover
ISBN: 978-3-319-27991-6