

REVIEW

FROM MONTE CARLO TO SCOTLAND YARD

Monte-Carlo Tree Search for Multi-Player Games

Pim Nijssen

PhD Thesis, Maastricht University 2013, 190 pp.¹

Reviewed by Dap Hartmann

Pim Nijssen's motivation to conduct the present research finds its roots in his childhood fascination with board games and computers. So when he discovered that there is a research field that combines the two, he was sold lock, stock, and barrel. Nijssen completed his bachelor degree in Knowledge Engineering and Computer Science at Maastricht University with the thesis 'Playing Othello Using Monte Carlo'. His master thesis in Artificial Intelligence was titled 'Using Intelligent Search Techniques to Play the Game Khet'. And in 2009 he began his PhD research at the Department of Knowledge Engineering at Maastricht University. The result is the present 190-page thesis 'Monte-Carlo Tree Search for Multi-Player Games' which he successfully defended on December 2, 2013.

The success of Monte-Carlo Tree Search (MCTS) in zero-sum two-player games is undeniable. It has progressed highly complex games such as Go beyond what was anticipated merely two decades ago. The question is, can this approach also be applied to multi-player games? Even though, as Nijssen writes, "applying MCTS to multi-player games is quite straightforward", it was not done until 2008 when Nathan Sturtevant applied it to Chinese Checkers. That was one year before Pim Nijssen started his PhD research, and so the obvious goal of his research is appropriately phrased in his Problem Statement: "How can Monte-Carlo Tree Search be improved to increase the performance in multi-player games?" This Problem Statement gives rise to four Research Questions, each of which is addressed and answered in a separate chapter in the thesis. These four chapters (4-7) are preceded by a general introduction (Chapter 1), an overview of search techniques and enhancements (Chapter 2), and an explanation of the four test domains used in the following three chapters (Chapter 3).

The thesis as a whole is very well structured, and so is each individual chapter. For example, Chapter 3 describes the four deterministic perfect-information games that are used as test beds to answer the first three Research Questions. For each game (Chinese Checkers, Focus, Rolit, and Blokus) there is a separate section and each section is subdivided into concise descriptions of the rules of the game, its complexity, and the domain knowledge. An instructive graphical representation of the complexity of 24 different games shows the staggering state-space complexity of Blokus (10^{279}). By comparison, Go 19x19 has a state-space complexity of 'only' 10^{171} . In contrast, the game state complexity of Go 19x19 (10^{361}) dwarfs the game state complexity of Blokus (10^{156}).

Chapter 4 addresses the first Research Question: "How can multi-player search policies be incorporated in MCTS?" Nijssen explores the application of \max^n , paranoid, and BRS in MCTS (appropriately named MCTS- \max^n , MCTS-paranoid, and MCTS-BRS) and tests how well these perform in the four multi-player games mentioned above. He also compares the results with minimax-based search techniques. The tentative conclusion is that \max^n performs best, mainly because the advantages that paranoid and BRS exhibit in minimax search do not apply in MCTS because there is no α - β pruning. Nijssen modified MCTS- \max^n so that it can prove positions and thereby play tactical lines better. He also applied the MCTS-Solver concept which was developed by Winands, Björnsson, and Saito. This configuration was tested by playing 3360 games of Focus, a sudden-death game (a game that may end abruptly by the creation of one of a pre-specified set of patterns). Nijssen showed that it is a genuine improvement. The detailed results of all experiments in this chapter are found in the 21 pages of Appendix B.

The selection phase is of crucial importance in MCTS because the selection phase determines how the tree is traversed until a leaf node is reached. Therefore, the second Research Question is "How can the selection phase of MCTS be enhanced in perfect-information multi-player games?" That is the topic of Chapter 5. Nijssen's answer is: by using Progressive History, a domain-independent enhancement for the selection phase of MCTS.

¹ The thesis can be downloaded from: https://project.dke.maastrichtuniversity.nl/games/files/phd/Nijssen_thesis.pdf

Progressive History combines Progressive Bias and the relative history heuristic. The performance of this enhancement was determined compared to the standard UCT selection strategy in Chinese Checkers, Focus, Rolit, and Blokus. The main conclusion is: “Depending on the game and the number of players, Progressive History wins approximately 60% to 80% of the games against MCTS without Progressive History [...]. With an increasing number of players, the performance of Progressive History drops, though it still remains a significant improvement over the standard UCT selection strategy.” These reservations show both the strength and the weakness of Nijssen’s research. It is admirable that he runs experiments on four different multi-player games with varying numbers of players, but it limits the depths of each individual experiment. And this makes it virtually impossible to reach general conclusions, as there are always exceptions in such a broad test bed. For example, Table 5.5 contains the win rates of the Progressive History heuristic against the Progressive AMAF (All Moves As First) heuristic. There are 11 entries in the table: Chinese Checkers (2, 3, 4, and 6-player variants), Focus (2, 3, and 4-player variants), Rolit (2, 3, and 4-player variants), and Blokus (which is a 4-player game by definition). The dominance of Progressive History over Progressive AMAF decreases with increasing number of players. For 6-player Chinese Checkers Progressive History actually performs a tiny bit worse, although that is not statistically significant. Had Nijssen only focussed on Chinese Chess and played more games at varying search depths, this effect might have been established more reliably, or it may have disappeared. The same is seen in Rolit: Progressive History in the 3-player variant of Rolit is less effective than in the 4-player variant. Again, it is not significant within the margins of error because of the small sample due to the broad scope of the experiments. The ultimate ‘getting lost in parameter space’ is exhibited by Table 5.1 where each of the 11 aforementioned configurations is evaluated for 10 different settings of the parameter W (a constant that determines the influence of Progressive History) which yields a whopping 110 individual results.

“How can the playouts of MCTS be enhanced in perfect-information multi-player games?” is the third Research Question and is addressed in Chapter 6. Nijssen applied x^n , paranoid, and BRS in two-ply searches during the playout phase, augmented with enhancements such as ϵ -greedy playouts (with a probability of ϵ , a move is chosen uniform randomly), move ordering, killer moves, k -best pruning and tree reusing. Several round-robin tournaments were played where each player used a different playout strategy. Due to time constraints, only two-ply search-based playouts were performed in the 3 and 4-player variants of Chinese Checkers and Focus. BRS showed the best performance but suffered from a reduction of the number of playouts per second. Nijssen: “Based on the experimental results we may conclude that search-based playouts for multi-player games may be beneficial if the players receive sufficient thinking time.”

The fourth and final Research Question “How can MCTS be adapted for hide-and-peek games?” is answered in Chapter 7. This chapter is different from the preceding ones as it deals with imperfect information. As test domain, Nijssen chose the game Scotland Yard, a hide-and-peek game created in 1983 which has the following properties: imperfect information for the seekers, asymmetry in the goals of the players, and cooperation between the seekers. Even though Scotland Yard can be played by three to six players, Nijssen focusses only on the 6-player version: one hider (Mister X) and five seekers. Scotland Yard is currently too complex for computers to solve. To handle the imperfect information, two different techniques were investigated: single-tree determinisation and separate-tree determinisation. In addition he developed Location Categorisation, a technique to bias the determinisation towards more likely positions. Because of the asymmetric nature of Scotland Yard, different domain knowledge must be used for the hider and the seekers. This leads to an interesting conclusion: “We found that, for the MCTS hider, it is best to assume during the playouts that the seekers do not know where the hider is, while the MCTS seekers perform best if they do assume where the hider is located.”

Finally, in Chapter 8 Nijssen summarises his conclusions and makes five recommendations for future research, such as applying other search policies (*e.g.*, Coalition-Mixer and MP-Mix), combining the Progressive History with other selection strategies (*e.g.*, Progressive Widening and Progressive Bias), and applying the results of the current research to other domains (popular two-player games such as Go and Hex, and modern multi-player board games such as Settlers of Catan and Carcassonne).

Much of this research was already published in several papers by Pim Nijssen and Mark Winands. It is a pleasure to see all of that now neatly combined into this first class thesis.