REVIEW

FROM TO αβ TO ABCD AND SMAB

Solving Games and All That
Abdallah Saffidine
PhD Thesis, Université Paris-Dauphine 2013, 175 pp.[1]
Reviewed by Dap Hartmann

As the foundation for this thesis, Abdallah Saffidine develops a framework for deterministic two-player games with perfect information and two outcomes, to represent best-first search (BFS) algorithms such as Proof Number search (PNS), Monte-Carlo Tree Search (MCTS) and the Product Propagation (PP) algorithm. “PNS is a best first search algorithm that enables to dynamically focus the search on the parts of the search tree that seem to be easier to solve”. PNS has been applied successfully in many games, especially in ‘difficult’ games such as Checkers, Shogi and Go. The problem with PNS is that it is resource-intensive because the entire game tree needs to be kept in memory. “The basic idea in MCTS is to evaluate whether a state *s* is favourable to *Max* via Monte Carlo playouts in the tree below *s*. A Monte Carlo playout is a random path of the tree below s ending in a terminal state.” MCTS has proven to be very successful in games such as Go where progress had been slow because of the large branching factor and the extensive horizon effects. Product Propagation was a relatively new concept to me. “PP is a way to backup probabilistic information in a two-player game tree search. It has been advocated as an alternative to minimaxing that does not exhibit the minimax pathology.” Minimax pathology, which was discovered independently by Dana Nau and Don Beal some 35 years ago, is the counter-intuitive effect that deeper minimax searches do not always result in better play. “PP was recently proposed as an algorithm to solve games, combining ideas from PNS and probabilistic reasoning.” Although the PP algorithm performs well in Go does not do so well in other games, such as Shogi. However, Saffidine shows examples of three games in which PP outperforms the more traditional search algorithms: The game of Y (a connection game invented by Claude Shannon), Domineering, and Nogo (a misère version of Go in which the first player to capture loses).

In Chapter 3, Saffidine adapts his framework to two-player games with multiple outcomes which results in a Best First Search (BFS) framework. Using a principled approach, he creates a ‘multi-outcome information scheme’ which he calls ‘multization’, not to be confused with the Multization app which stands for ‘Multiplication X Memorization’, a fancy version of multiplication tables. Saffidine uses multization to generalize PNS and PP for multi-outcome games. The resulting Multiple-Outcome Proof Number Search (MOPNS) algorithm is applied to two games: Connect Four and Woodpush. Although Connect Four was already solved in 1988 by Victor Allis and James Allen, it still provides an interesting benchmark to test search algorithms. For 89% of the 256 4x5 positions that were tested, MOPNS needed fewer nodes than PNS but at the expense of requiring 16% more time. The same pattern was found for the 625 5x5 positions that were tested and in which MOPNS needed fewer nodes in 65% of the cases using 14% more time. For Woodpush, a recent game that involves forbidden repetition of the same position, the results were comparable to the performance exhibited in Connect Four: fewer nodes at the price of more time.

Chapter 4 investigates the relationship between Multi-agent Modal Logic K (MMLK) and sequential game search and suggests several new model checking algorithms. Saffidine shows how the MMLK Model Checking framework can be used to develop new research in game tree search. He focusses on turn-based games with perfect and complete information. Not just two-player games, such as Chess and Go, but also single-player games (puzzles) like Rubik’s Cube and Sokoban, and multiplayer games such as Chinese Checkers (a variation of the game Halma that is played by two, three, four or six people). Saffidine suggests several new model checking algorithms for MMLK and proves that one of them (Minimal Proof Search - MPS) is correct and that, under certain conditions, it minimizes a generic cost function. However, MPS has some limitations because it is a memory-intensive best-first search algorithm that currently cannot make use of transpositions.

The final chapter deals with two-player zero-sum games with simultaneous moves, the so-called stacked-matrix games. For this domain, Saffidine developed the Simultaneous Move Alpha-Beta (SMAB) algorithm which is a generalized version of the alpha-beta pruning algorithm and is described as “a depth-first search algorithm

[1] This thesis can be downloaded from:
http://cgi.cse.unsw.edu.au/~abdallahs/Papers/2013/Solving%20Games%20and%20All%20That.pdf

[which loops] through all joint action pairs first checking trivial exit conditions and if these fail, proceeding with computing optimistic and pessimistic bounds for the entry in questions, and then recursively computing the entry value.” SMAB involves a large computational overhead because it needs to solve Linear Programs. Saffidine developed heuristical optimizations to speed up this process. The result was an algorithm that solved Goofspiel faster than alternative methods (backward induction and sequence form solver). Goofspiel (also known as The Game of Pure Strategy or GOPS) is a card game in which three of the four suits are used and the players move simultaneously. Saffidine experimented with various numbers of cards per suit to analyze the pruning efficiency as a function of game-tree size.

Saffidine also discusses the Alpha-Beta (Considering Durations) algorithm (ABCD), an efficient heuristic algorithm for Real-Time Strategy games which involve simultaneous moves under tight time constraints. One such game is Starcraft which, according to Wikipedia, “[m]any of the industry's journalists have praised […] as one of the best and most important video games of all time, and for having raised the bar for developing real-time strategy games”. Unfortunately, I am totally ignorant of this type of game, so I will merely cite Saffidine on his future goal in this domain: “Our next steps will be to integrate ABCD search into a STARCRAFT AI competition entry to gauge its performance against previous year’s participants, to refine our combat model if needed, to add opponent modelling and best-response-ABCD to counter inferred opponent combat policies, and then to tackle more complex combat scenarios.”

‘Solving Games and All That’ is an excellent thesis with a solid game-theoretical framework that uses 42 definitions, 8 theorems, 36 propositions, 1 lemma, 11 algorithms, 15 examples and 7 remarks, distributed over 5 chapters in the space of 175 pages. It is well structured and written in a smooth style that reads like a charm. The definitions and descriptions of both well-known concepts and new notions that Saffidine provides are exemplary. So much so that I have liberally quoted from his text because I cannot phrase many of Saffidine’s descriptions better myself.

Overview of Playing Hall in Yokohama.