Improving Game-tree Search by Incorporating Error Propagation and Social Orientations

(Extended Abstract)

Brandon Wilson
University of Maryland
Department of Computer Science
College Park, Maryland 20742
bswilson@cs.umd.edu

ABSTRACT

Game-tree search algorithms, such as the two-player Minimax algorithm and its multi-player counterpart, Max-n, are a fundamental component in the development of computer programs for playing extensive-form games. The success of these algorithms is limited by the underlying assumptions on which they are built. For example, it is traditionally assumed that deeper search always produces better decisions and also that search procedures can assume all players are selfish and ignore social orientations. Deviations from these assumptions can occur in real games and can affect the success of a traditional search algorithms. The goal of my thesis is to determine when such deviations occur and modify the search procedure to correct the errors that are introduced.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Graph and tree search strategies*

General Terms

Economics, Algorithms

Keywords

game-tree search, multi-player games, heuristic search

1. INTRODUCTION

Game-tree search algorithms, such as the two-player Minimax [5] algorithm and its multi-player counterpart, Max-n, are a fundamental component in the development of computer programs for playing extensive-form games. In fact, game-tree search algorithms have contributed greatly to the success of computerized players in two-player games, producing players that are as good or better than the best human players [6].

Despite this success, these algorithms are limited by the underlying assumptions they are built upon. My work focuses on two of these assumptions: 1) deeper search produces better, more informed decisions and 2) players are rational agents that are indifferent to their opponents' utility.

My first problem focuses on the generally accepted belief that deeper search results in better game-play. In the early 1980s, however, Nau [3] discovered a class of games that exhibits a phenomenon known as *game-tree pathology*, in which deeper minimax search results in worse performance. Mutchler [2] later discovered that pathology also exists in the multi-player adversarial search algorithm, max-n. More recently, game-tree pathology has been shown to exist in two chess endgames and kalah [4]. My goal is to develop a method for recognizing the portions of a game-tree that introduce pathological behavior and then to dynamically adjust search depth in these portions of the search to improve decision accuracy.

My second problem concerns the importance of inter-player relationships in multi-player games. For example, consider a game in which a player has lost all "practical" chances of winning, but still can influence the outcome of the game. The typical approach to dealing with this problem has been to make simplifying assumptions. Max-n, for example, assumes that all players are rational and do not consider other players' utilities. The Paranoid algorithm [7], on the other hand, assumes that while the searching player attempts to maximize its own utility, all other players have formed a coalition against that player. In the real world, these assumptions often do not hold. In fact, relationships can change drastically throughout a single game as the circumstances change. The goal of this work is to develop a way to explicitly capture, learn, and utilize these social preferences in the search procedure.

2. ERROR MINIMIZING SEARCH

In pathological game trees, searching deeper is less likely to produce a move with maximal utility. Most games such as chess, checkers, and the like have been thought to not be pathological: deeper searching minimax algorithms tend to result in better play. As such, little work has been focused on game-tree pathology since its discovery.

However, it has recently been shown that even non-pathological games, such as chess, exhibit *locally-pathological* characteristics [4] where portions of the search can reduce decision accuracy despite an improvement in overall accuracy. Therefore, the work in this section is intended to formally define and characterize the notion of error in game-tree search, leverage it to identify local pathologies, and improve decision

Cite as: Improving Game-tree Search by Incorporating Error Propagation and Social Orientations (Extended Abstract), Brandon Wilson, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1355-1356.

accuracy in games with any degree of local pathology.

2.1 Progress to Date

We initially focused on two-player games and the problem of defining error with respect to a game tree. We examined a simplified representation of a game tree and static evaluation function.We identified probabilistic rules for propagating error based on the type of node (i.e., forced-win, forced-loss, or critical node) in the tree. Integrating this error calculation with the minimax search procedure forms what we refer to as Error Minimizing Minimax (EMM). The algorithm propagates both minimax values and error values simultaneously, replacing the propagated value with the static evaluation when the propagated error exceeds the static evaluation error. Similarly, we developed a multi-player algorithm, Error Minimizing Max-n (EMMN), for multi-player games.

Initial experimental results on a board-splitting game indicate improvement over classical minimax [9] and max-n search. Neither EMM nor EMMN exhibit pathological behavior in the same circumstances that induce such behavior for minimax and max-n.

2.2 Future Directions

The next step with this work is to apply it to real games. Specifically, endgame chess and kalah, which were shown to have situations that are pathological [4], would be a significant step for this work. Applying our error minimizing search to real games requires that we estimate the error associated with a static evaluator. This is significantly more difficult than in the case of the board-splitting game since completely solving such games is not possible. Therefore, correlating the evaluation with the true minimax value is not possible. One potential solution is a Monte-Carlo approach but we will need to evaluate this and other potential solutions empirically.

3. SOCIALLY ORIENTED SEARCH

Unlike two-player games, where interpersonal relationships are unlikely to arise, interpersonal relationships can have a significant effect on the outcome of a multi-player game; some games even have interpersonal relationships as an integral component to success (e.g., Settlers of Catan and Diplomacy). Incorporating these relationships into the heuristic function directly is the only solution we have seen for this in the literature. There are two problems with this approach: 1) heuristic functions are already difficult to design, requiring vast amounts of domain knowledge for a strong estimate and 2) the heuristic function is typically designed offline and before the game is played, so once the game is started, the relationships cannot be altered unless other evaluation functions have been prepared and can be swapped.

Our goal with this work is to represent social preferences of one's opponents, learn these preferences as the game progresses, and successfully integrate the preferences into the game-tree search. This model of social preferences will complete the concept of an opponent model in multi-player games where much work has already been done to model individual evaluation functions [8].

3.1 Progress to Date

Our work is built upon a recently suggested social-range matrix model [1] of social preferences that supports the description of interpersonal orientations as captured in the social behavior spectrum. The social matrix construct makes it possible to model "socially heterogeneous" systems where players may have different social orientations toward each of the other players.

We incorporate the social-range matrix into a search we refer to as Socially Oriented Search (SOS). We use the player's social orientation to transform each evaluation vector to be a linear combination of each player's utility. Then we estimate the social matrix by simply averaging the effects of each player's recent move history. For example, a player that tends to make moves that negatively affect player i's state and positively affect player j's state will be seen as cooperating with player j and competing with player i. This generalization allows the SOS algorithm to implement both Max-n and Paranoid algorithms, as well as an infinite number of other possibilities, by simply modifying the social-range matrix.

We empirically evaluated the SOS algorithm in the four-player game Quoridor against opponents with random preferences. SOS significantly outperformed two multi-player game-tree search algorithms (Max-n and Paranoid).

3.2 Future Directions

The next step in this work is to experiment with more robust learning algorithms for learning the social-range matrix. Our goal with learning the social matrix is twofold: 1) learn the social-range matrix as accurately as possible and 2) learn it quickly and be able to account for relationship changes that occur abruptly during the game. There is a tradeoff in that having more data (i.e., a longer move history) improves the chances of inferring accurate relationships and at the same time if these relationships are dynamically changing then this information can quickly become stale.

4. REFERENCES

[1] P. Kuznetsov and S. Schmid. Towards Network Games with Social Preferences. *Structural Information and Communication Complexity*, pages 14–28, 2010.

[2] D. Mutchler. The multi-player version of minimax displays game-tree pathology. *Artificial Intelligence*, 64(2):323–336, 1993.

[3] D. S. Nau. An investigation of the causes of pathology in games. *Artificial Intelligence*, 19(3):257–278, 1982.

[4] D. S. Nau, M. Luštrek, A. Parker, I. Bratko, and M. Gams. When is it better not to look ahead? *Artificial Intelligence*, 174(16-17):1323–1338, 2010.

[5] J. V. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[6] J. Schaeffer, J. Culberson, N. Treloar, B. Knight, P. Lu, and D. Szafron. A world championship caliber checkers program. *Artificial Intelligence*, 53:53–2, 1992.

[7] N. R. Sturtevant and R. Korf. On pruning techniques for multi-player games. In *AAAI*, pages 201–208, 2000.

[8] N. R. Sturtevant, M. Zinkevich, and M. H. Bowling. Prob-maxn: Playing n-player games with opponent models. In *AAAI*, pages 1057–1063, 2006.

[9] B. Wilson, A. Parker, and D. S. Nau. Error minimizing minimax: Avoiding search pathology in game trees. In *Proceedings of International Symposium on Combinatorial Search (SoCS-09)*, July 2009.