

## **Part A**

# **Cognitive Game Theory: Beyond Alpha-Beta Search**

By

**Jeremie Pouly, Jennifer Novosad, and Justin Fox**

Covering

**Alpha-Beta and Its Extensions; An Evolutionary Algorithm Applied to Chess;  
Inductive Adversary Modeler**

## **Part B**

Gross, R. K. Albrecht, W. Kantschik, W. Banzhaf. *Evolving Chess Playing Programs*.  
Proceedings of the Genetic and Evolutionary Computation Conference. 2002.  
(<http://citeseer.ist.psu.edu/gro02evolving.html>)

Walczak, Steven. *Knowledge-Based Search in Competitive Domains*. IEEE  
Transactions on Knowledge and Data Engineering, Vol. 15, No. 3, May/June  
2003.  
(<http://64.233.161.104/search?q=cache:wZnG1v8GAmcJ:carbon.cudenver.edu/~swalczak/tkde03.pdf+AI+opponent+modeling+to+prune+alpha+beta+search+trees+IAM&hl=en>)

## **Part C**

Games have long been a test bed for innovative artificial intelligence algorithms. In our lecture, we begin with a discussion of one of the most basic and pervasive game-playing algorithms, the alpha-beta minimax search, highlighting a number of its more recent adaptations including transposition tables, heuristic move order selection, and variable depth searching. Using a checkers game programmed by our team, we demonstrate the power of this basic search. We next proceed to describe a recent application of genetic programming and evolutionary algorithms to the construction and optimization of alpha-beta chess-playing programs. This distributed system, known as goopy, was available for home users to download from the internet and help evolve chess programs with their spare CPU power. The players evolved could defeat some of the best chess-playing algorithms available while searching only 50% of the nodes, demonstrating the power of genetic computing. Finally, we present Steven Walczak's Inductive Adversary Modeler (IAM), which incorporates a model of the opponent's strategy to predict likely moves and prune the search tree. IAM allows for a 12.5% increase in search depth when used against chess grand masters. It has been extended to other gaming domains, and with some more complicated changes to the model of opponent moves, could be extended to other competitive domains like finance and military tactics.

## **Part D**

Banzhaf, W., P. Nordin, R.E. Keller, and F.D. Francone. *Genetic Programming – An Introduction On The Automatic Evolution of Computer Programs and its Applications*. 1998.

This paper, originally published in German, describes the concepts behind genetic algorithms, the methods of crossover and mutation, and other such important background information. It is helpful for understanding the portion of our lecture devoted to evolving computer chess players.

Schaeffer, J. *The History Heuristic and Alpha-Beta Search Enhancements in Practice*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(11):1203-1212, 1989.

A good background paper for those just starting to learn about chess programming. The paper contains information on the basic alpha-beta search design as well as several heuristics for improving upon it.

Schwefel, H. P. *Evolution and Optimum Seeking*. John Wiley and Sons, Inc., 1996.

A more general overview of evolutionary algorithms and their implementation. Readers wishing to delve more deeply into aspects of evolutionary algorithms would find this helpful.

Walczak, Steven. *Improving Opening Book Performance Through Modeling of Chess Opponents*, ACM 0-89791-828-2/96/02. Philadelphia PA 1996.

This paper modifies IAM to work specifically on opening books. It predicts about 65% of the moves played by chess masters in the opening. Readers interested in how to model the beginning of a game would find this helpful.

## **Part E**

The project was designed to divide more or less evenly among three people. We all will be meeting frequently to discuss ideas and the various stages of the project, but we each also have our specific spheres of focus. Jeremie will be responsible for creating our demonstration and the introduction to our lecture. Namely, he will be quickly describing the basic alpha-beta game tree search algorithm and demonstrating it with some form of checkers game to perhaps be played by a student. Jennifer will be speaking about algorithms which model the adversary's strategies and moves, and use them to prune the search tree. Justin will be talking about evolutionary algorithms, in particular a method by which a naturally-selected alpha-beta chess-playing algorithm was created

using a massively distributed framework. Each of us plans to speak for about 25 minutes, leaving a few minutes for overflow or questions. We plan to create our slides together, so that our presentation flows smoothly. Each person will then most likely be responsible for annotating his or her own slides. We believe we have divided the work as evenly as possible, and we will continue to meet together to ensure that no one feels overburdened and that the project is truly a group effort.

## **Part F**

As mentioned above, Jeremie will be creating our demonstration. He plans to use the skeleton of an alpha-beta checkers playing algorithm Justin coded for fun after 16.413. He will be adapting this basic barebones algorithm and attempting to institute some of the more advanced search techniques talked about in the literature we cite. We hope to implement some subset of the following: transposition tables, heuristic move ordering, search deepening on interesting branches, quiescence search, advanced heuristic functions, and perhaps others.

Jeremie will begin our lecture by introducing the basic alpha-beta algorithm. Then he will demonstrate the workings of this simple algorithm by perhaps allowing a student to play against the computer or else detailing a previously played game. At the end of our talk, he will then present his updates to the simple alpha-beta search and allow students to match their wits with the program with whatever time remains.