Computer Systems Project Proposal
An Othello Artificial Intelligence Machine Learning

Nick Sidawy, Period 5

October 31, 2006

1 Problem Statement

To create an Othello artificial intelligence that uses machine learning to maximize its playing ability.

2 Purpose

The purpose of this research project is to implement machine learning with artificial intelligence. The reason why I chose this project is two-fold. First, to create a very effective Othello AI for anyone's enjoyment. Second, and more acedemically oriented, is to gain a deeper understanding of machine learning, a subject which I am very interested in. Anyone who is interested in having a significant Othello challenge will be interested in the results of this project.

3 Scope of Study

The first thing I would like to achieve with this project is to program an effective foward-checker for my AI. The second goal is to use a genetic algorithm to formulate the best evaluation function for the AI to use. My last goal is to have the AI learn from each move it makes so that the more it plays, the faster and better it will perform

4 Background and review of current literature/research in this area

Machine learning is an extensive field of study. Most of what is done with machine learning is tied to artificial intelligence which is why Othello seemed to be a good vessel for my research. It is simple enough game for me to work on, yet difficult enough to keep me working throughout the quarters. Obviously machine learning is not limited to board games. For example, recently at MIT, students created a small robot child with an artificial intelligence. The goal of the project was to implement machine learning so the robot would be able to teach itself to walk using trial and error; it ended up working very well.

Although this seems as if it is miles away from an Othello artificial intelligence, it is closer than one might think. A student last year used machine learning (genetic algorithms) in order to create the best evaluation function. As I have stated, I would like to do something similar to this but also go a step further by having it learn from the opponent's moves. Most of what I use and learn to complete this project will come from the CSL's Artificial Intelligence book.

The ”state of the art” in the field of board game artificial intelligence is Chessmaster. It is a chess artificial intelligence that is so skilled that it was able to beat the international chessmaster Larry Christiansen in 2002.

5 Procedure and Methodology

Most of this project will be coded in java because it is fast and contains many of the tools that will be necessary for my project, such as timers and GUI's that are simple to create and use. The program will be tested two different ways. The first way is to test it against human players. The second way will be to test the AI against previous AI's to see if it has improved or worsened after changes have been made. Margins of victory (or loss) along with running time will determine how well it is performing. Since making the AI better or faster usually requires rather small changes yet clever in the code, it is very easy to work on certain segments from week to week. The project is broken up into smaller parts which can be worked on without affecting other components which makes tweaking one part very easy.

The only true way to test for bugs in this sort of AI is to play against it.

This is rather frustrating at times because one either has to play enough for a specific situation to arise or recode the program so the situation comes up on the start-up, which is most closely related to "Structural and Functional" testing. In rare instances I am able to use outputs in order to look for bugs, for example, returning the evaluation of the move chosen and making sure that this was indeed the best move to make.

The speed of the program is exponentially related to the ply, the amount of moves it searches ahead, of the artificial intelligence ($O(n^p)$ where p is the ply). This is the only mathematical formula that I can envision being critical for testing my program. It is important to realize the substantial toll that making the AI slightly better has on the runtime.

By the end of the quarter, I hope my AI is able to search at least six moves ahead with little to no delay. In order to do this I will need to make a forward-checker that makes use of Alpha-Beta pruning (a way to eliminate possibilities before they come up) in a very effective way. I must also make sure that all simple methods that are used to make the actual Othello game work, such as capturing pieces, work in the most efficient way possible.

The two main algorithms that I will work on during this quarter are the forawad-checking and evalutaion methods. The way the foward-checker works is by looking through each possible move that the computer has, making one of the moves, looks at the moves the opponent is presented with, and repeats. In other words, the goal is to traverse a tree of possible moves and picking the move that will lead to the best scenario down the line. The ply determines how many levels of the tree it goes through. The trick about this algorithm is that at each level of the tree it picks the move that is best for which player it is simulating for. Therefore, the computer assumes that its opponent will play perfectly. This is why it has been labeled the Minimax algorithm. First it looks for the move that is best for it, then at the opponent's move that will be worst for it. (See Diagram A)

The other algorithm that will be worked on is the evalution function. This function returns a number rating how good a specific scenario is for a player. It does this based on the positions of the pieces and amount of available moves for each player. For example, pieces in the corners are very valuable so they will add many more points to the rating then a piece near the center.

Creating visuals from the results of my project is rather difficult because I am not getting a specific output each time I run the program. I suppose the best way to visualize my progress would be to graph the results of simulated games

The second and third quarters will mainly be devoted to the second and third goals, respectively, mentioned in the scopes section. The third goal will most likely be the most time consuming and will be worked on during the second quarter as well as the third quarter.

6 Expected Results and Value to Others

When my project is complete I simply plan on having a very good Othello AI that uses machine learning extensively and will be challenging to the best of players. Furthermore, I expect to have a much deeper knowledge of machine learning. The final results will be presented in an Othello GUI that I plan to post on my TJHSST website for anyone to play.