# Othello Artificial Intelligence with Machine Learning

Computer Systems TJHSST 2006-2007
Nick Sidawy

## Introduction

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 complex enough to give the project significant depth.

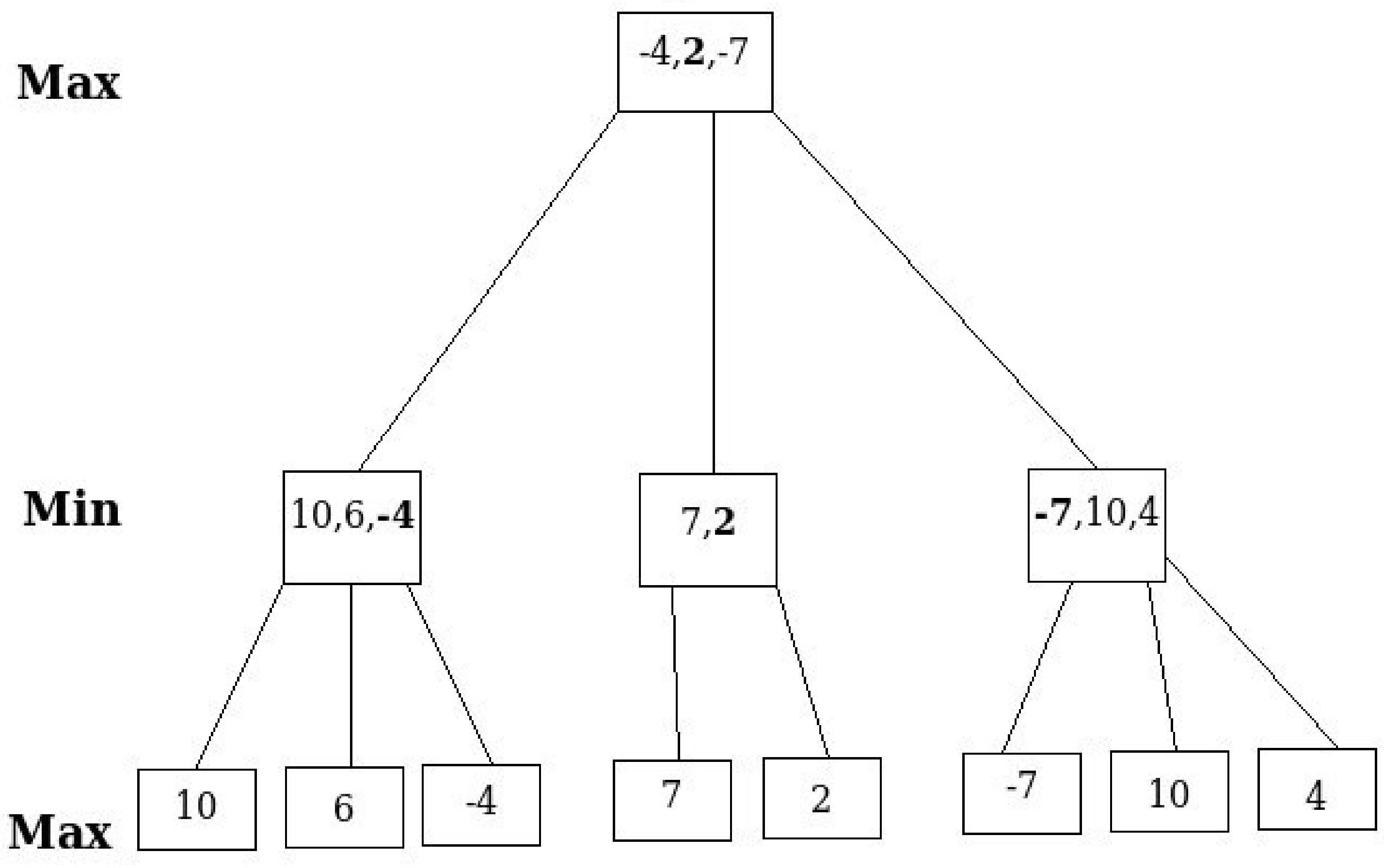
Obviously machine learning is not limited to board games. For example, recently at MIT, students created a small robot child with an artificial intelligence. The goal of the project was to implement machine learning so the robot would be able to teach itself to walk using trial and error.

Although the robot child seems as if it is miles away from an Othello artificial intelligence, it is closer than one might think.

## Abstract

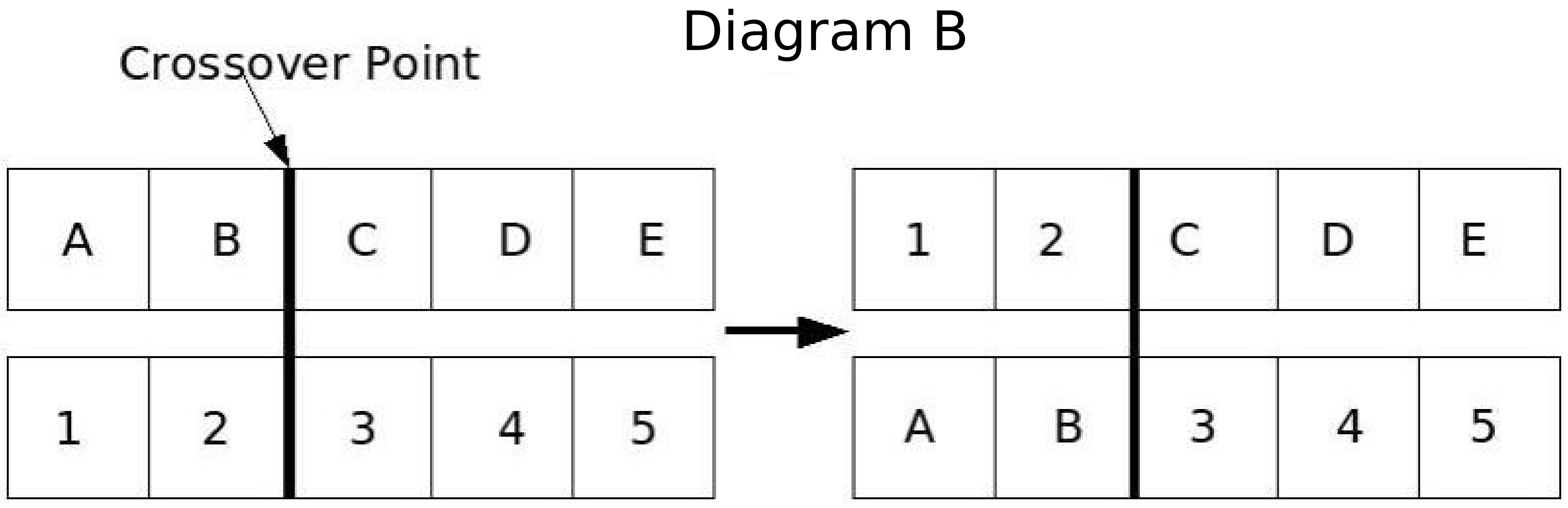
The purpose of this research project is to implement machine learning with artificial intelligence. The reason why I chose this project is two-fold. First, to create an effective Othello AI to play and have fun with. Second, and more academically oriented, is to gain a deeper understanding of machine learning, a subject which I am interested in.

Diagram A



This is an example of a three ply Minimax tree. As you can see it traverses three levels and then begins to return evaluation ratings. At the second level it looks at all the evaluation ratings returned from the third level and returns the minimum (shown in bold). At the first level it looks at these selected values and returns the maximum (shown in bold).

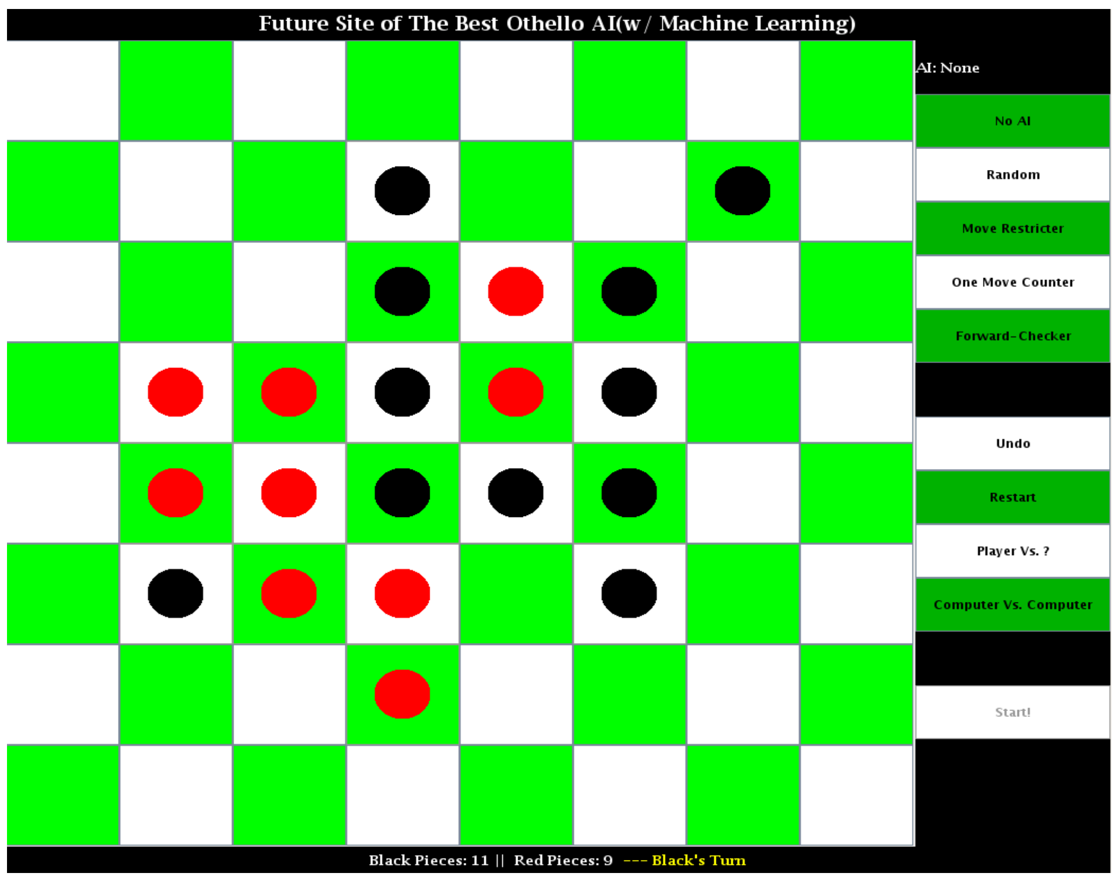
Diagram B



## Procedure

There are two main algorithms that are used in this project, the first of which is the forward-checker. The goal of the forward checker is to find the best possible move. 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 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. (See Diagram A)

The other algorithm is the evaluation 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.


Future Site of The Best Othello AI(w/ Machine Learning)

AI: None

Black Pieces: 11 || Red Pieces: 9 --- Black's Turn

To find the best evaluation function I used the Genetic Algorithm(GA). The GA is a sort of Darwinian evolution cycle that tests the quality of different evaluation functions and then splices the strongest ones together (as shown in Diagram B) to produce the next generation (a new, presumably stronger, set of functions). The next generation will go through the same process and eventually an optimal evaluation function will be reached.

In order to increase the efficiency of the minimax algorithm and effectiveness of the AI, the AI will learn from each move it makes in a game. Every move will be recorded, along with the board and its evaluation, into a HashMap. Therefore, the next time a similar board situation comes up, all the AI will need to do is look into the HashMap for the move and this will save the time and trouble of traversing through the entire tree of possible moves.

## Results

The culmination of the project is an Othello AI that uses machine learning extensively and is a challenge for skilled players.