

6364-Artificial Intelligence

---

# Othello game agent design

---

April 26, 2017

Net ID: 161430

RENCHI LIU

UT-Dallas

## Table of Contents

<b>Abstract .....</b>	<b>2</b>
<b>Introduction .....</b>	<b>2</b>
<b>Related work.....</b>	<b>3</b>
<b>Approach .....</b>	<b>3</b>
<b>Implementation .....</b>	<b>3</b>
<b>Experiments and results .....</b>	<b>4</b>
<b>Conclusions/discussion.....</b>	<b>5</b>
Using the Greedy algorithm:.....	5
Using the Min-Max algorithm and Alpha-Beta algorithm: .....	5
Improve aspects: .....	5
<b>References .....</b>	<b>6</b>

## Abstract

Othello is a strategy board game for two players, played on an 8×8 board. The object of the game is to have the majority of disks turned to display your color when the last playable empty square is filled.<sup>[1]</sup>

Othello is a typical adversarial search problem. The Othello game agent was created for gaining more practical knowledge about this problem. It is aim to play with human being, and beat them. In order to win the game, different intelligent level agents will have different algorithms based on one evaluate function to guide them in proceeding steps. The algorithms include Greedy algorithm , Min-Max algorithm and Alpha-Beta algorithm .

Below the report, there will be a detailed analyze about different algorithms and some future prospective. Last, there will be an evaluation about different algorithms .

## Introduction

In March 2016, AlphaGo beat Lee Sedol in a five-game match, the first time a computer Go program has beaten a 9-dan professional without handicaps.<sup>[2]</sup> Artificial intelligence shocked people all over the world again. One of the key reason AlphaGo can beat human is that it can predict future situations longer than human brain in the game board.

Under some certain rules, computer can search every possibility and find the best solution to react. This agent has one evaluate function to evaluate each possible position. The agent will use algorithms to try each position ,and simulate future board situation about next possible steps. This way helps the agent to evaluate and choose current step reasonable. In conclusion, the key point of the AI is how to find the best steps based on current situation and future possibility.

There will be three algorithms and one evaluate function to helps the agent to make a decision. The evaluate function tells whether the position is good or not ,according to some common sense. For example, the corner is the best position to put disks, because the opponent can not traverse one in the corner . The three algorithms are Greedy algorithm , Min-Max algorithm and Alpha-Beta algorithm . Greedy algorithm don't use the evaluate function , it only focus on how to reverse more disks. Min-Max algorithm and Alpha-Beta algorithm will use evaluate function to calculate positions score in order to get the best moving position.

## Related work

Paul S. Rosenbloom from Department of Computer Science, Carnegie-Mellon University who published "A World-Championship-Level Othello Program<sup>[3]</sup>". In this research, he created an agent to take part in the Tournament and also made a deeper analysis about the evaluation function and different algorithms. Also include more statistical analysis about time and running process.

Michael Buro from NEC research institute who published "An Evaluation Function for Othello Based on Statistics<sup>[4]</sup>". In this research, he focus on the time and potential movability in the evaluation function.

## Approach

- (1) An agent can play Othello with human, make decision based on the future situation .
- (2) Agent can use three different kinds of algorithms to play this game.
- (3) The difference between using an evaluation function and non- evaluation function .
- (4) The difference in running time between Min-Max algorithm and Alpha-Beta algorithm.

## Implementation

This program consists of UI-display part, game-play part and AI-agent part.

In the UI-display part:

Using pygame (3-rd game lib) to build UI and proceed the user action.

In the game-play part:

Checking the steps are legal or not , the available reverse and handle the reverse. After one player finish the steps and change the board. It will send data to the UI-display part in order to update the UI.

In the AI-agent part:

AI can make a reasonable step decision, after implemented evaluation function and algorithms.

## Experiments and results

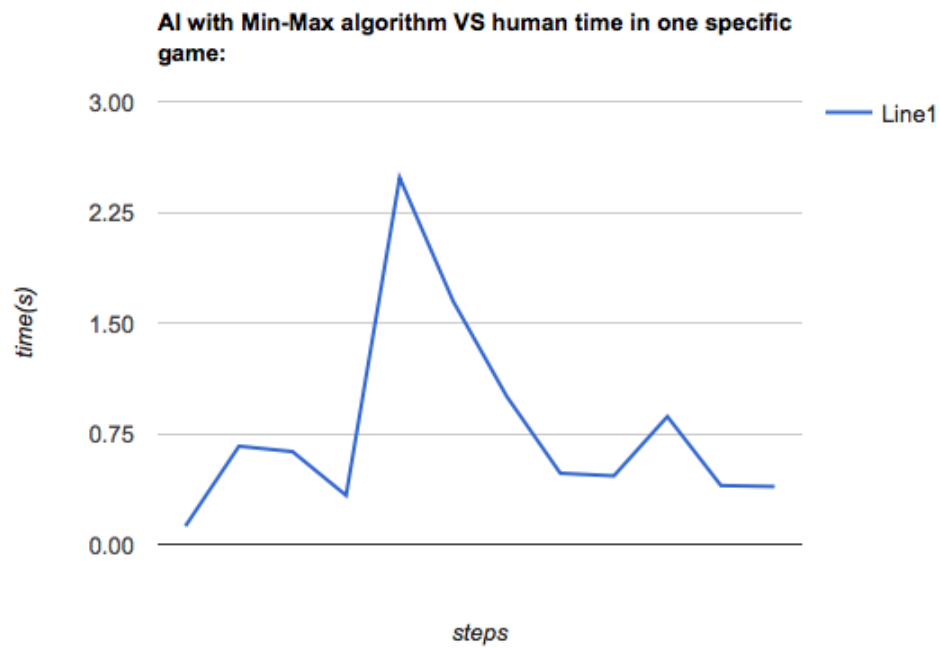
**Result** of AI with Greedy algorithm (no evaluation function) VS human:

time \ result	1	2	3	4	5	6	7	8	9	10
win	√	√	x	√	x	x	√	x	x	x
lost	x	x	√	x	√	√	x	√	√	√

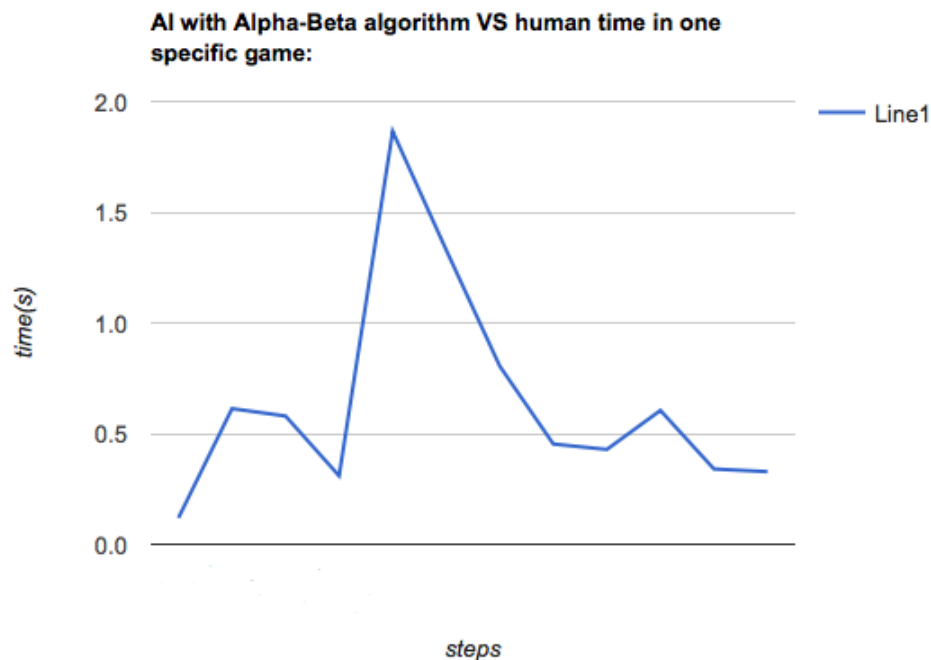
**Result** of AI with Min-Max algorithm VS human:

time \ result	1	2	3	4	5	6	7	8	9	10
win	√	√	√	√	x	x	√	√	√	x
lost	x	x	x	x	√	√	x	x	x	√

**Time** of AI with Min-Max algorithm VS human time in one specific game:



Time of AI with Alpha-Beta algorithm VS human time in one specific game:



## Conclusions/discussion

Using the Greedy algorithm:

The agent's reaction is fast, but only focus on current situation. It cannot beat human often. What's more, it's reaction can be predict by human ,because human can easily understand the Greedy algorithm and learn how to find the pattern to defeat the agent. The result is the Greedy algorithm is a simple and week strategy to build AI agent. We need more powerful way.

Using the Min-Max algorithm and Alpha-Beta algorithm:

Min-Max algorithm and Alpha-Beta use the same evaluation function. The difference between the two algorithm is the second one has the prune function . So it can save much time in the search Tree . As a result, we can see even the decision the two algorithms made are the same . But the time they used are huge different. This mean the prune function actually improved the efficiency in Alpha-Beta algorithm.

Improve aspects:

The evaluation function only use a simple rule to determine the value of different position in board. But as the game is playing, the value of the position will be different. Even

the corner and side are good position . But the agent need merge more reality situation. So the evaluation function can be improve, like the evaluating rule can be change in the game according to current situation .

Also , the agent can add mobility evaluation in the decision part. Because, the mobility can also influence the search space. And influence the decision. The agent should try to create more mobility chance in the game. Not only focus on the position. More information can be found in Michael Buro's published "An Evaluation Function for Othello Based on Statistics".

## References

[1]. <https://en.wikipedia.org/wiki/Reversi>

[2]. <https://en.wikipedia.org/wiki/AlphaGo>

[3]. <http://www.sciencedirect.com/science/article/pii/0004370282900030>

[4]. <https://skatgame.net/mburo/ps/evalfunc.pdf>