Isolation Game Heuristics Report

Matin Moezzi

July 19, 2020

I implemented my own custom agent using the minimax algorithm with two different heuristics and improved it with the alpha-beta pruning and iterative deepening. I tested my own custom agent against the minimax opponent with baseline heuristic and the following results were obtained.

Heuristic 1. Offensive to Defensive

This heuristic consists of both the offensive and defensive approach. In the first half of the game, the agent attempts to minimum opponent legal moves but inversely in the second, the agent legal moves become important. In other words, in the first half we will play offensively and in the second defensively. We emphasize the legal moves with a weighted linear function. Mathematically it can be expressed as:

$$EVAL(s) = \begin{cases} (p \times 2) - o, & \text{if } m \leq 0.5 \\ p - (o \times 2), & \text{if } m > 0.5 \end{cases}$$

Where m is the ratio of total moves to the board size, o and p is the number of the opponent and the player legal moves respectively.

Heuristic 2. Block to Opponent[1]

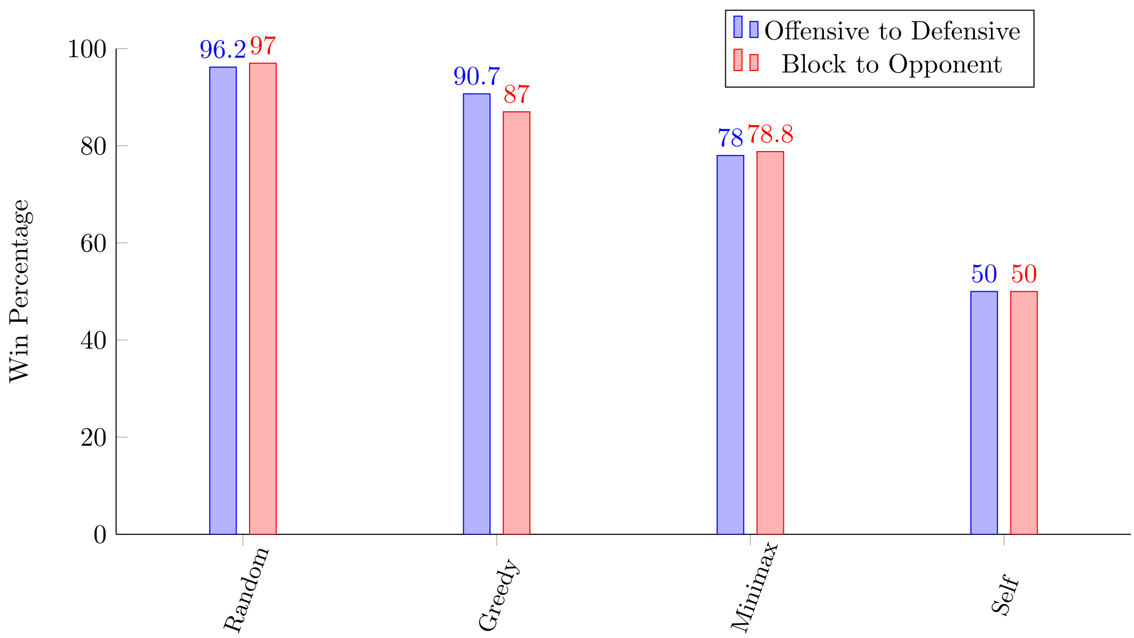
This heuristic is completely aggressive and seeks to hunt the opponent. To carry it out we need to create a list of possible moves for our agent (N_p) and the opponent (N_o), then create an array of those movements that are equal and use them in our calculation.

$$\mathbf{A}_{i,j} = \sum_{i=1}^{N_p} \sum_{j=1}^{N_o} \text{if } (p_i == o_j) \text{ then } p_i$$

$$EVAL(s) = p - (o \times 2) + size(\mathbf{A})$$

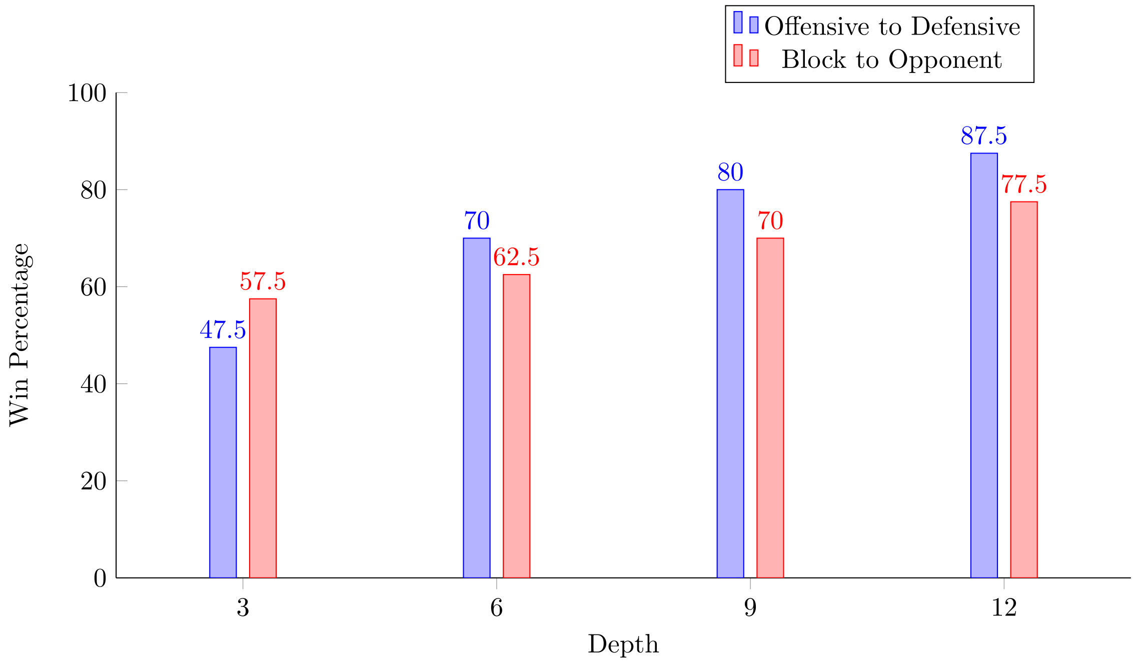
Baseline Heuristic. #my-moves - #opponent-moves

Figure 1: Win Percentage of my own custom agent in 150 rounds



I also tested my own custom agent against the baseline minimax agent through the different depths in 20 rounds. Results are shown in the bellow as:

Figure 2: Win Percentage through different depth vs Baseline Minimax



1. **What features of the game does your heuristic incorporate, and why do you think those features matter in evaluating states during search?**

   Both implemented heuristics applied offensive and defensive strategies. The defensive game attempts to expand states that have more legal moves for the player but in the offensive game, the player tries to the opponent has fewer legal moves. *Offensive to Defensive (OTD)* heuristic has the advantage of switching between offensive and defensive games. In other words, *OTD* is a dynamic heuristic according to the game board situation. In *Block to Opponent (BTO)* heuristic, in addition to the legal moves of both players, the number of the same action of two players is an important factor. In this way, the opponent's legal decisions reduced because the available states are common with the player states and occupied with the player's moves.

2. **Analyze the search depth your agent achieves using your custom heuristic. Does search speed matter more or less than accuracy to the performance of your heuristic?**

   As you can see in Figure 2, the depth of the search algorithm is a very important factor and performance of the agent increased by increasing in depth. In bounded time game, speed is an important aspect of the agent performance because it must to decide which action is optimal in a specific state at an appropriate time. Therefore the agent speed, as well as its accuracy, is a very crucial factor in the agent performance. Adding search depth increases agent accuracy but reduces search time as well. In contrast, the shallow depth search reduces accuracy but has good speed. So, it is a trade-off between speed and accuracy.

Reference

[1] Search Heuristics for Isolation, Julio Cesar Aguilar Jimenez, January, 2018