Heuristic Analysis

Joao Paulo Fernandes Ventura

The tournarment test consists of rounds of 5 matches each with a 150 miliseconds timeout. The results were compared to AB_Improved as baseline.

The x-axis correspond chosen custom heuristics:

1. AB_Improved
2. AB_Custom
3. AB_Custom_2
4. AB_Custom_3

Given `opponent` to the the number of possible moves to the adversary player, and `player` the number or moves of my agent, the custom heuristics score are given by:

1. `player - 1.618033988749895*opponent`
2. `player - 2*opponent`
3. `player/opponent`

Ten tournaments were executed, with 5 matches each, obtaining the following win rate percentual:

	win rate (% mean)	Win rate (% std)
AB_Improved	75.57	2.55
AB_Custom	77.43	3.68
AB_Custom_2	78.14	3.56
AB_Custom_3	78.14	3.86

Table 1. Average win rate

One might say that AB_Custom_2 and `AB_Custom_3` are the winner heuristics, but in fact all algorithms are even.

As we can see in the candestick chart, all measurements overlap if we consider each standard deviation measurement, thus due the current sampling number if is not possible to define a winner strategy.

Future improvements to all strategies we should consider to enlarge the gap between them to `AB_Improved` are:

- Use multiprocessing or multithreading to calculate each branch of the moves possibility tree in parallel.
- Create opening moves, such as we previously did on `AIND-Sudoku`

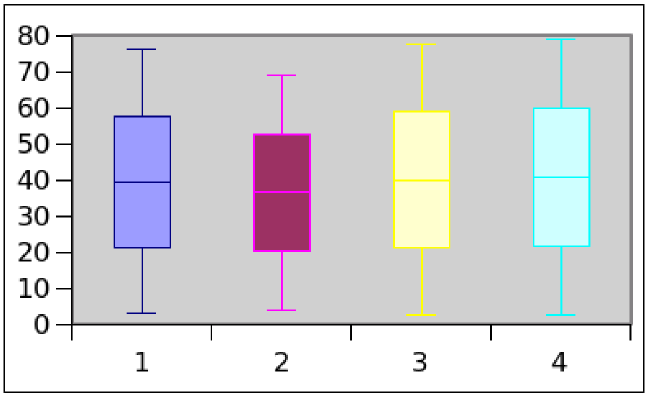

Figure 1. Average win rate with standard deviation

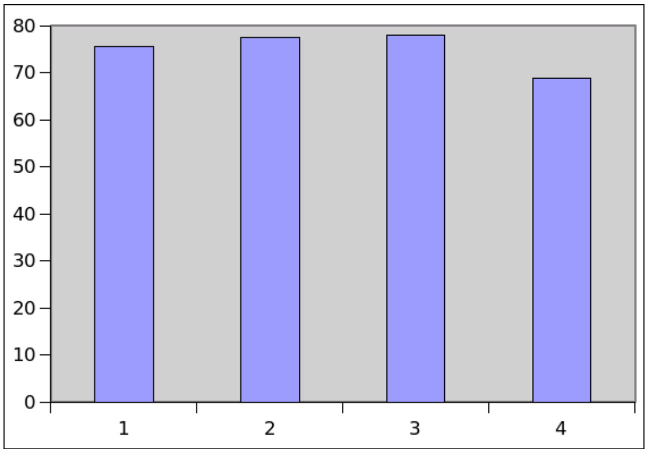

Figure 2. Average win rate

We also should run a least 1024 tournaments in order to the measurements be statistically relevant, reducing the standard deviation, keeping in mind that 10 tournaments consumed 01h53min59s or real time.