Heuristics Analysis

Lucas Cabral Rezende

custom_score()

This heuristics subtracts the square of the opponent’s legal moves from the agent’s legal moves, so it gives more emphasis on reducing the ability of the opponents to move and could be seen as a more aggressive heuristics.

custom_score_2()

This heuristics subtracts the square of the opponent’s legal moves from the square of the agent’s legal moves, as both scores can grow at the same rate a search method using this heuristics will give preference to moves that enhance the ability of our agent to move in the future.

custom_score_3()

This heuristics subtracts the cubic of the opponent’s legal moves from the square of the agent’s legal moves, like custom_score() it gives more emphasis on reducing the ability of the opponents to move, but it also tries to find moves where our agent has a lot of options to move, being a sort of hybrid of custom_score() and custom_score_2().

Conclusion

On the tests I have run with tournament.py custom_score() yield the best results, about 75% winning rate on average, demonstrating that among my heuristics the best alternative is to always reduce our opponents future options.