Ultimate tic tac toe report

Szymon Kosakowski

October 2020

1 Introduction

This report shows results of testing agents on the ultimate tic tac toe game. Agents used in the experiment were: Flat MC, MCTS, MAST, NST. The code was written in Python 3. All the test were performed offline on my own machine.

2 Flat MC and MCTS

At first I have measured the average number of simulations per second for both agents.

Table 1: Average number of simulations per second

	in first turn	in whole game
Flat MC	212	273
MCTS	169	254

Then agents were tested against random agent with the same time budget of 1 second. There were 100 of games played for both agents. Flat MC got 100% wins and MCTS 99% wins. As this data doesn't say much besides that both agents probably work fine, then they were tested against each other also with the same time budget of 1 second and outs of 100 games MCTS won 24 games, Flat MC - 76.

3 Parameters of MAST and NST

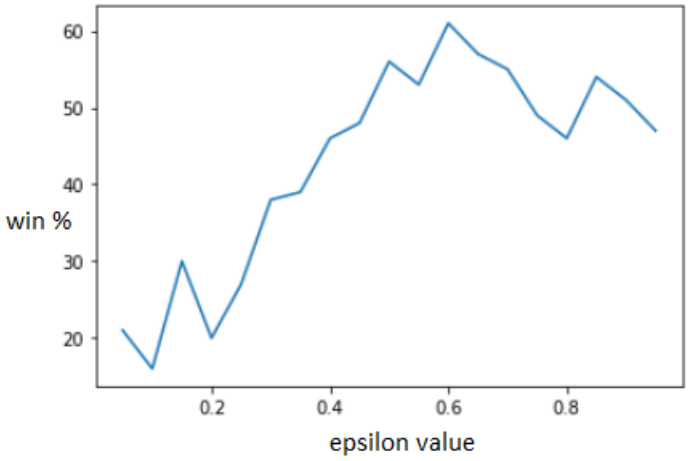
Before tuning the parameters I checked number of simulations per second of both enhancements.

Table 2: Average number of simulations per second

	in first turn	in whole game
MAST	88	115
NST	124	237

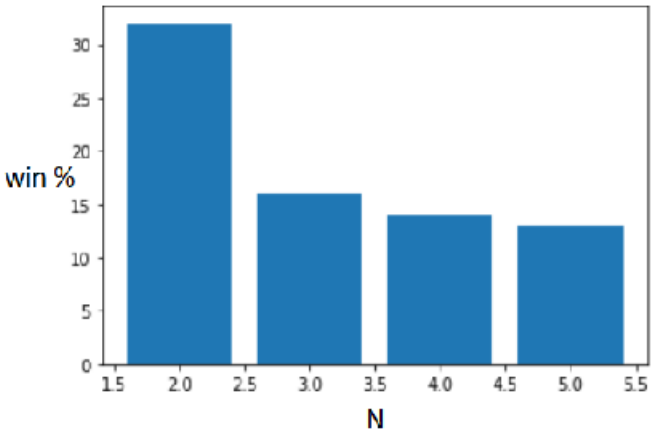
MAST and NST agents were playing 100 games for each value of the parameter: ϵ for MAST, N for NST with MCTS as reference agent. Best *epsilon* for MAST was chosen as the best ϵ in NST also.

In MAST testing I started testing ϵ at 0.05 and added 0.05 every 100 games.



$\epsilon_{best} = 0.6$ was chosen as best.

Then as mentioned before N parameter of NST was tested with this ϵ_{best}



According to the test $N_{best} = 2$ was chosen.

4 Method Comparison

After tuning the parameters all agents played 100 games against each other, again with the same time budget of 1 second.
The table presents win % of each agent.

Table 3: Agents win ratio

	Flat MC	MCTS	MAST	NST	random agent
Flat MC	-	76%	79%	81%	100%

Table 4: Agents win ratio

	Flat MC	MCTS	MAST	NST	random agent
MCTS	24%	-	39%	68%	99%

Table 5: Agents win ratio

	Flat MC	MCTS	MAST	NST	random agent
MAST	21%	61%	-	40%	93%

Table 6: Agents win ratio

	Flat MC	MCTS	MAST	NST	random agent
NST	19%	32%	60%	-	92%

In conclusion the tests shown that Flat MC algorithm is best for this task. MCTS and the enhancements are not as effective. It is probably because low amount of games per second in every method, but Flat MC has the most out of all. Maybe it would be a better idea to write the code for this experiment in other, faster programming language.