Fast and Furious Game Playing : Monte Carlo Drift Specifications report

Prateek Bhatnagar, Baptiste Bignon,
Mikaïl Demirdelen, Gabriel Prevosto,
Dan Seeruttun--Marie, Benoît Viguier

Supervisors: Nikolaos Parlavantzas, Christian Raymond

11/27/2014

The MCTS algorithm has been chosen in order to develop our artificial intelligence. However, in order to improve its efficiency, we will need to adapt it. The main problem is the branching factor[1] of the Arimaa game which average is 17 281 and reaches about 22 000 after 10 moves[2].

Game	Average number of possible moves
Othello	8
Chess	35
Game of Go	250
Arimaa	17 281

The reason why the branching factor of a game is so important is because it increases greatly the space that has to be searched in order to guess what will happend multiples moves ahead. In chess after 6 moves, the number of positions evaluated are about 35^6 which is roughtly equivalent to 1,8 billions. In Arimaa, after 3 turns (yours, the opponent and yours again), if you were the explore all positions, you would need to evaluate around 5,2 trillions[3] boards (2000 times more than chess with half the number of moves).

In order to decrease the space to be search, our MCTS Algorithm will perform a big number of simulations before chosing the nodes to explore. After the selection, it will prune the tree in order to optimise search speed and the memory managment.

[1] In a tree, the branching factor is the number of children at each node.
[2] http://arimaa.janzert.com/bf_study/
[3] 1 trillion = 1 thousand billions = 10^{12}.