[11pt]article fullpage graphicx mathtools

Adversarial search of Monte-Carlo simulation for 3D Tic-Tac-Toe

Kohei Kawasaki

document sec:rw Introduction

Brief Description of problem As the game is getting more complex and deeper strategically, the heuristic function and cost function is getting useless due to absence of absolute evaluation of current state and forward step. That node will be expanded exponentially and it could be said impossible to use A* search or IDA* since both algorithms depends on the validity of evaluation function and strategically complexity causes evaluating a each move to difficult. In alpha-beta pruning, if we put bounds on the possible values of the utility function, the nwe can arrive at bounds for the average without looking at every number. Therefore Monte-Carlo simulation(Tree search) works an alternative evaluation of Alpah-Beta prunning or other search algorithm. From a start position the algorithm play thousands of games against itself, using randomly chosen move and evaluate each note by the statical win percentage. Monte Carlo Tree Search MCTS, does not rely on a positonal evaluation function, however this approach is a general algorithm and can be applied to mamy problems. The most promising result was the game of Go. In this project, I am going to work on 3D tic-tac-toe from the approach of monte carlo tree search. 3D tic-tac-toe is the three dimensional version of commonly played board game. The strategy of game expanded due the expansion of dimension. Assume N as the length of each side, the size of the board game N^3.$Thusthemaximumtreesizeistheorem N^3!$.$Intermsonlysizeofboardgame, from 5sidelength, itisoverthegameofgo.Howeverbe$
$tac-toe.Hereisthecategorisofproblems : (1)Singleplayerploblems(calledoptimizationproblems), (2)twoopponentproblems,$
$tac-toeisalsonon-cooperativetwoplayerdeterministicgame.Monte-CarloTreeSearchMonteCarloTreeSearchisthebestf$

table Different types of problems. tabular— l —— c — c — c — One player Two player Multi Player