ADVERSERIAL SEARCH (21STNOV 2019)

Pick an Experiment

Select at least one of the following to implement and evaluate in your report. (There is no upper limit on the techniques you incorporate into your agent.)

Option 1: Develop a custom heuristic (must not be one of the heuristics from lectures, and cannot only be a combination of the number of liberties available to each agent)

Option 2: Develop an opening book (must span at least depth 4 of the search tree)

Option 3: Build an agent using advanced search techniques (for example: killer heuristic, principle variation search (not in lecture), or monte carlo tree search (not in lecture))

Choice

Option 3: Build an agent using advanced search techniques namely monte carlo tree search(MCTS).

Reasons:

1. MCTS was not taught in lecture, so could take effort to learn it by myself and implement it.
2. MCTS process is conceptually simple.
3. MCTS is popular, even my cat knew it. 😊

Baseline: The alpha-beta search with iterative deepening from the classroom has been considered as the baseline

TABLE 1. Comparison of Iterative deepening alpha-beta search with MCTS

(Fair Matches: False)

Search technique	Search time	Matches	Depth(ABS) / Budget(MCTS)	Matches won against Minimax Agent .
Alpha-beta search	150	10	3	25%
Alpha-beta search	1000	10	5	40%
Alpha-beta search	150	100	3	27%
Alpha-beta search	1000	100	5	34%
Monte Carlo Tree Search,	150	10	35	80%
Monte Carlo Tree Search,	1000	10	75	55%
Monte Carlo Tree Search,	150	100	25	58%
Monte Carlo Tree Search,	1000	100	75	66%

TABLE 2. Comparison of Iterative deepening alpha-beta search with MCTS

(Fair Matches: True)

Search technique	Search time	Matches	Depth(ABS) / Budget(MCTS)	Matches won against Minimax Agent .
Alpha-beta search	150	10	3	37.5%
Alpha-beta search	1000	10	5	32.5%
Alpha-beta search	150	100	3	Unable to run after 2 tries.
Alpha-beta search	1000	100	5	24%
Monte Carlo Tree Search,	150	10	35	60%
Monte Carlo Tree Search,	1000	10	75	67.5%
Monte Carlo Tree Search,	150	100	25	56%
Monte Carlo Tree Search,	1000	100	75	70%

MONTE CARLO TREE SEARCH

- Choose a baseline search algorithm for comparison - Alpha-beta search with iterative deepening. How much performance difference does your agent show compared to the baseline?
    - The baseline algorithm gives a success rate of around 30-40%.
    - MCTS search almost doubles the success rate to around 60-70%.
    - Fair matches = true makes the result more consistent in MCTS.
- Why do you think the technique you chose was more (or less) effective than the baseline?
    - MCTS does not depend on the choice of a heuristic. Selecting a heuristic needs knowledge of the search domain which are not required in this case.
    - MCTS backpropagates the outcome of each game immediately which ensures all values are always upto-date following every iteration of the algorithm. This allows the algorithm to return an action from the root at any moment in time; allowing the algorithm to run for additional iterations often improves the result.
    - The tree selection allows the algorithm to favour more promising nodes (without allowing the selection probability of the other nodes to converge to zero), leading to an asymmetric tree over time. In other words, the building of the partial tree is skewed towards more promising and thus more important regions.

(Reference - A Survey of Monte Carlo Tree Search Methods)

mcts-survey-master.pdf

LIMITATIONS

The project submission has the following limitations:

1. Unable to select the chosen search method by passing a parameter.
2. Code is not organized and is procedural.
3. Commenting of the code will be completed in the later revision.
4. Submission is late due to other commitments.
5. Testing took major part of the time and has been done as comprehensively as possible.