adverserial Search (17th Nov 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. Not taught in lecture so could learn it by myself and implement it.
2. MCTS process is conceptually very simple
3. MCTS is popular, even my cat knew it. 😊

Baseline: The alpha-beta search code 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** | **Budget/Depth** | **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 |  |
|  |  |  |  |  |

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

(Fair Matches: True)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Search technique** | **Search time** | **Matches** | **Budget/Depth** | **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 |  |
| Alpha-beta search | 1000 | 100 | 5 |  |
| Monte Carlo Tree Search, | 150 | 10 | 35 | 60% |
| Monte Carlo Tree Search, | 1000 | 10 | 75 | 67.5% |
| Monte Carlo Tree Search, | 150 | 100 | 25 |  |
| Monte Carlo Tree Search, | 1000 | 100 | 75 |  |
|  |  |  |  |  |

# Uninformed search

The depth first graph search where the nodes are expanded to the deepest level of the search tree has the least new node expansions against the number of actions in the domain. It might not be optimal for complex ones with its increased plan length.

The breadth first graph search where the nodes are expanded at a given depth at a time has more new node expansions against the number of actions in the domain. It seems a better alternative for complex problems requiring less time to any other uninformed search.

The uniform cost search is similar to the breadth first search when all path costs are equal. It might be useful when the path cost of each cost is variable. In the four air cargo problems, if the cost of moving from between airports is variable then the uniform cost search will give the

In the current case the breadth first graph search seems the optimal uninformed search in terms of plan length, time and number of nodes expanded.

# Informed search

All the informed searches give the same plan length for the problems 1 & 2.

* Greedy best first graph search

The greedy best first graph search with unmet goals heuristics shows the best performance in terms of time taken for problems 1 & 2 which is validated while running it for problem 3 & 4.

The levelsum heuristics give about similar performance for the greedy best first graph search for the four problems with levelsum being slightly better.

The performance of the greedy best first graph search with setlevel heuristics is the least optimal, especially in the case of complex problems.

* A\* search

The A\* search with unmet goals heuristics shows the best performance for A\* in terms of time taken for problems 1 & 2 which is validated while running it for problem 3 & 4.

The levelsum heuristics result in better performance for the A\* search for the four problems then the maxlevel and setlevel heuristics.

The maxlevel heuristic causes longer runtime for A\* search than greedy best first graph search.

The performance of the A\* search with setlevel heuristics is the least optimal, especially in the case of complex problems, which result in longer runtimes. (Tried running it but it is still running after around 4+ hours)

* Which algorithm or algorithms would be most appropriate for planning in a very restricted domain (i.e., one that has only a few actions) and needs to operate in real time?
  + Greedy best first graph search with unmet goals.
* Which algorithm or algorithms would be most appropriate for planning in very large domains (e.g., planning delivery routes for all UPS drivers in the U.S. on a given day)
  + Greedy best first graph search with unmet goals.
* Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?
  + Greedy best first graph search with unmet goals.

TABLE 1. Actions, plan length, new node expansions and time taken by problems 3 & 4.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Search type** | **Heuristics** | **Actions in the domain/Plan Length** | | **New node expansions** | | **Time taken in seconds.** | |
| Problem 3 | Problem 4 | Problem 3 | Problem 4 | Problem 3 | Problem 4 |
| Breadth first search | NA | 88/12 | 104/14 | 14663 | 99736 | 1.99 | 6.11 |
| Depth first graph search | NA | 88/392 | 104/ 24132 | 408 | 25174 | 0.594 | 1168.01 |
| Uniform cost search | NA | 88/12 | 104/14 | 18510 | 113339 | 3.268 | 10.494 |
| Greedy best first graph search | Unmet goals | 88/15 | 104/18 | 25 | 29 | 0.0184 | 0.0479 |
| Greedy best first graph search | Levelsum | 88/14 | 104/17 | 14 | 17 | 5.466 | 4.9673 |
| Greedy best first graph search | Maxlevel | 88/13 | 104/17 | 21 | 56 | 3.95 | 8.187 |
| Greedy best first graph search | Setlevel | 88/17 | 104/23 | 35 | 107 | 29.94 | 63.1 |
| A-star search | Unmet goals | 88/12 | 104/14 | 7388 | 34330 | 3.455 | 6.1885 |
| A-star search | Levelsum | 88/12 | 104/15 | 369 | 1208 | 75.87 | 194.76 |
| A-star search | Maxlevel | 88/12 | 104/14 | 9580 | 62077 | 1348.88 | 8169.44 |
| A-star search | Setlevel | 88/12 | XX/XX | 3423 | XX | 1718.15 | XX |
|  |  |  |  |  |  |  |  |