ASEN 6519-007 Decision Making under Uncertainty
Homework 3: Online MDP Methods

February 12, 2020

1 Conceptual Questions

Question 1. (20 pts) Consider an MDP with three states, $|\mathcal{S}| = 3$ and two actions, $|\mathcal{A}| = 2$. Draw the following three trees to a depth of $d = 2$, with circles representing state nodes and squares representing action nodes:

(a) The complete state-action tree produced with forward search.

(b) A sparse sampling tree with $n = 2$.

(c) A partial tree after 4 iterations of Monte Carlo tree search (according to Algorithm 4.9).

Question 2. (30 pts) In the proof for Lemma 5 of the Sparse Sampling paper by Kearns, Mansour, and Ng, (`https://www.cis.upenn.edu/~mkearns/papers/sparsesampling-journal.pdf`), the authors claim that if a policy π satisfies $|Q^*(s, \pi^*(s)) - Q^*(s, \pi(s))| \leq \beta$ for all $s \in \mathcal{S}$, then it immediately follows that $|R(s, \pi^*(s)) - R(s, \pi(s))| \leq \beta|$. This is a mistaken conclusion. Provide an MDP and a policy that presents a counterexample to this claim and demonstrate that the statement does not hold.

2 Challenge Problem

Question 3. (50 pts) Your task is to create an online planner for randomly generated 100x100 grid world problems. In these grid world problems there is a reward of +100 every 20 cells, i.e. at [20,20], [20,40], [40,20], etc. Once the agent reaches one of these reward cells, the problem terminates. All cells also have a randomly generated cost.

For each of the 100 randomly generated grid worlds that the planner is evaluated on, you will have 50ms of offline solve time, and for each online step, you will have 50 ms to make a decision of what action to take. If the time limits are exceeded, random actions will be taken. You can create one sampled grid world with `DMUStudent.HW3.DenseGridWorld()` and examine it using the POMDPs.jl interface. The submission should be a `POMDPs.Solver` such as an `MCTSSolver` or `POMDPs.Policy`.

There are no restrictions for this exercise, except that you may not attempt to deliberately exceed the timing or hack any other part of the evaluation script. You may use any packages, multi-threading, etc. In particular, the MCTS.jl package is suitable for solving this problem. A score of 50 or more will receive full credit.