# 15-887 Solutions Homework 3

## 1 Execution

### 1.1

Correct answer: c

a) is false. For a new $\epsilon$ it may be the case that $f(s_{goal}) <=$ minimum $f$-value in $OPEN$. In that case, no states are expanded.

b) is false. The whole point of ARA* is to reduce the number of expansions when compared to the execution of multiple weighted A*.

c) is true. Refer to a). In this case, the solution for the new $\epsilon$ is the same as the one for the previous $\epsilon$.

d) is false. ARA* may still be able to reuse g-values.

### 1.2

We have to prove that the updated heuristic $h^u$ is still consistent.

We start by proving that $h^u(G) = 0$. LRTA* only updates the heuristic when a node is expanded. The moment the agent reaches the goal, the search is stopped. Therefore, $h^u(G)$ remains 0.

We now have to prove that for all $s$ and its sucessors $s'$, $h^u(s) \leq c(s, s') + h^u(s')$. Without loss of generality, assume we start by updating a single state $s$. By the definition of the LRTA* update we have:

$$h^u(s) = \min_{s'} \{c(s, s') + h(s')\} \leq c(s, s') + h(s') = c(s, s') + h^u(s'),$$

for all $s'$ successor of $s$. Note that we assumed we were only updating state $s$ for now, and because of that $h^u(s') = h(s')$.

Finally, we have to prove that when changing the value of the heuristic at $s$ we still have $h^u(s') \leq c(s', s) + h^u(s)$:

$$h^u(s') = h(s') \leq c(s', s) + h(s) \leq c(s', s) + h^u(s),$$

where we used the fact that the original heuristic was consistent to begin with.

Thus, we proved that after the update of $s$, the heuristic $h^u$ remains consistent. By induction, we can successively apply this reasoning to the next steps and guarantee the consistency of the final heuristic.

## 2 Policy Iteration

### a)

$|\mathcal{A}|^{|S|} = 3^3 = 27.$

### b)

Adapted from the answer by Devin Schwab.

Start with the random policy $\pi_0$, which selects the *offensive* action in all states.

We now compute the value of this policy (policy evaluation), following the equation:

$$V_\pi(s) = R(s) + \gamma \sum_{s' \in S} T(s, \pi(s), s')\, V_\pi(s').$$

There are three states, and thus three equations with three unknowns:

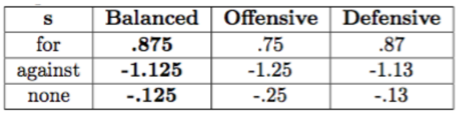
$$\begin{aligned}
V_{\pi_0}(\text{for}) &= R(for) + 0.5 \sum_{s' \in S} T(\text{for}, \text{offensive}, s')\, V_{\pi_0}(s') \\
&= 1 + .5\left[.25 V_{\pi_0}(\text{for}) + .5 V_{\pi_0}(\text{against}) + .25 V_{\pi_0}(\text{none})\right] \\
V_{\pi_0}(\text{against}) &= -1 + .5\left[.25 V_{\pi_0}(\text{for}) + .5 V_{\pi_0}(\text{against}) + .25 V_{\pi_0}(\text{none})\right] \\
V_{\pi_0}(\text{none}) &= 0 + .5\left[.25 V_{\pi_0}(\text{for}) + .5 V_{\pi_0}(\text{against}) + .25 V_{\pi_0}(\text{none})\right]
\end{aligned}$$

Solving this system yields
$V_{\pi_0}(\text{for}) = .75$
$V_{\pi_0}(\text{against}) = -1.25$
$V_{\pi_0}(\text{none}) = -.25$

We then perform policy improvement using the formula

$$\pi_1(s) = \arg\max_{a \in \mathcal{A}} R(s) + \gamma \sum_{s' \in S} T(s, a, s')\, V_{\pi_0}(s')$$

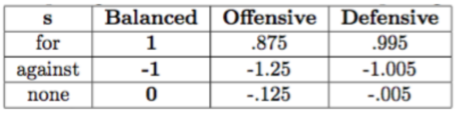
The table below describes the values for each state-action pair. For each state, the best action is bolded. Therefore, the new policy becomes *balanced* for all states.



| **s** | **Balanced** | **Offensive** | **Defensive** |
|---|---|---|---|
| for | **.875** | .75 | .87 |
| against | **-1.125** | -1.25 | -1.13 |
| none | **-.125** | -.25 | -.13 |

Performing policy evaluation we arrive at
$V_{\pi_1}(\text{for}) = 1$
$V_{\pi_1}(\text{against}) = -1$
$V_{\pi_1}(\text{none}) = 0$

We perform policy improvement again, and get the following state-action pairs values:



| **s** | **Balanced** | **Offensive** | **Defensive** |
|---|---|---|---|
| for | **1** | .875 | .995 |
| against | **-1** | -1.25 | -1.005 |
| none | **0** | -.125 | -.005 |

We arrived at the same policy: *balanced* for all states. Therefore, policy iteration finishes with the optimal policy being *balanced* for all states.

### c)

Typically, different discount factors may lead to different optimal policies. However, for this particular example, that is not the case.

Note that, since none of the transition probabilities are affected by the starting state, different discount factors will affect the state values, but not their relative orderings. In fact, the relative ordering will only be influenced by the immediate rewards of the states.

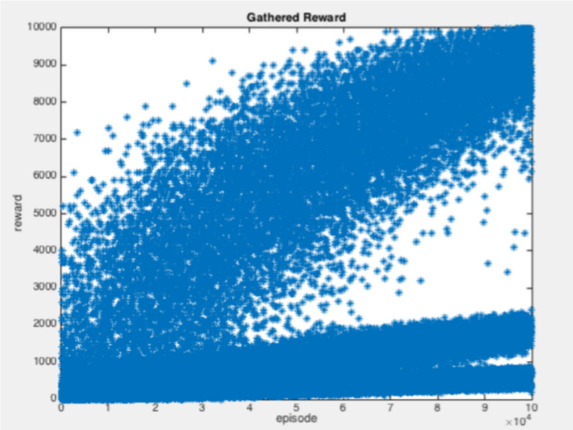
With this in mind, and since the reward also only depends on the state, we will see the same phenomenon during the policy improvement step. Specifically, the state-action pairs values may be different, but the ordering will be the same.

Because of that, for this particular problem, having different discount factors will not lead to different optimal policies.

## 3 Q-Learning

The goal of this exercise was to properly implement the $Q$-learning algorithm, while using a reasonable exploration strategy, and possibly dynamic learning rate. Reasonable exploration strategies include $\epsilon$-greedy, where $\epsilon$ decreases throughout learning, or the the Boltzmann strategy.

Using 100,000 episodes with 100 iterations each, a plot with the total summed reward per episode should look similar to the plot below.



Credits to Kainan Peng.

## 4 Policy Reuse

The $\pi$-reuse strategy contributes with a similarity metric between policies through the concept of the gains $W$. The larger the $W$, the more similar two policies are.

Intuitively, if reusing a past policy $i$ leads to larger rewards when learning a new policy $j$, then $i$ and $j$ should be similar.

## 5 Experience Graphs

Correct answer: d

a) is false. As seen in class, the experiences can be disconnected.

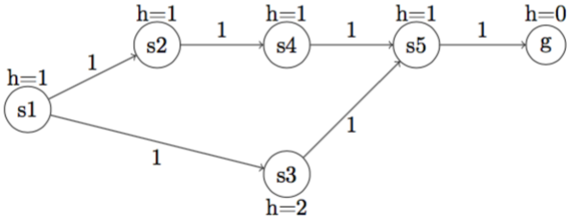
b) is false. $h^E$ is always $\epsilon^E$ consistent, independently of the quality of the experiences. This comes from the definition of $h^E$.

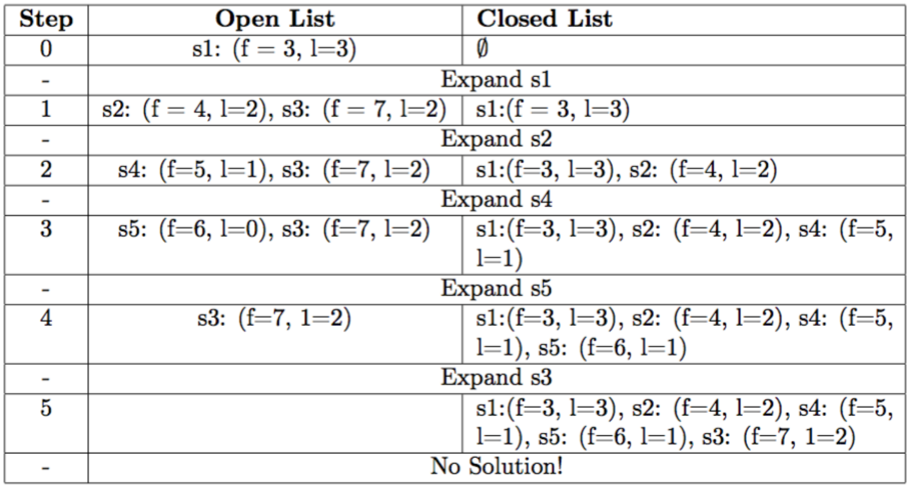
c) is false. Even if all the experiences provided are optimal, there are no optimality guarantees. For example, the experiences may all lie in a region of the graph very far away from the start and goal positions. In that case, the experiences may never be used.

# 6 Dependent vs. Independent Variables

Credits to Devin Schwab.

Assume initial battery level $l = 3$, and $\epsilon = 3$. The problem is that the inflation leads us to expand $s5$ from $s4$ and not from $s3$. Thus, when we finally expand $s3$, $s5$ is already in the closed list. Because of that, we fail to find the solution $s1, s3, s5, g$.





| Step | Open List | Closed List |
|---|---|---|
| 0 | s1: (f = 3, l=3) | ∅ |
| - | Expand s1 | |
| 1 | s2: (f = 4, l=2), s3: (f = 7, l=2) | s1:(f = 3, l=3) |
| - | Expand s2 | |
| 2 | s4: (f=5, l=1), s3: (f=7, l=2) | s1:(f=3, l=3), s2: (f=4, l=2) |
| - | Expand s4 | |
| 3 | s5: (f=6, l=0), s3: (f=7, l=2) | s1:(f=3, l=3), s2: (f=4, l=2), s4: (f=5, l=1) |
| - | Expand s5 | |
| 4 | s3: (f=7, 1=2) | s1:(f=3, l=3), s2: (f=4, l=2), s4: (f=5, l=1), s5: (f=6, l=1) |
| - | Expand s3 | |
| 5 | | s1:(f=3, l=3), s2: (f=4, l=2), s4: (f=5, l=1), s5: (f=6, l=1), s3: (f=7, 1=2) |
| - | No Solution! | |