Assignments (/assignment/) / Homework 5

Home (/assignment/hw5/)

Problem (/assignment/hw5/problem/)

Actions

Homework 5

Problem

Messing with Rewards

Note: More information forthcoming.

We saw in the lesson that regardless of the potential-based shaping function given, the optimal policy will remain unchanged. The purpose of choosing a particular shaping function, however, is to try to speed learning. In this question, we will explore how to pick a potential-based shaping function to speed up the convergence of policy iteration (PI).

Consider the following MDP:

MDP 1

```
int numStates = 9;
int numActions = 4;
double[][][] probabilitiesOfTransitions = {{\{1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0\},{1.
0,0.0,0.0,0.0,0.0, \{0.0,0.9,0.0,0.0,0.1,0.0,0.0,0.0,0.0\}, \{0.0,0.1,0.1,0.0,0.8,0.0,0.0\}
0},\{0.0,0.8,0.1,0.0,0.0,0.1,0.0,0.0,0.0\},\{0.0,0.1,0.1,0.0,0.0,0.8,0.0,0.0,0.0\}},\{\{0.0,0.1,0.1,0.0,0.0,0.0,0.0\}
\{0.8,0.0,0.1,0.0\},\{0.0,0.8,0.0,0.1,0.0,0.1,0.0,0.0,0.0\},\{0.0,0.1,0.0,0.8,0.0,0.0,0.0\}
0,0.0,0.0,0.1,0.1,0.0,0.0,0.8},{\{0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0\},\{0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0\}}
.0,0.0\},\{0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0\},\{0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0\}\},
,0.0,0.0,0.0,0.1,0.0,0.8,0.1},{0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.1,0.9}}};
0,0.0}, {{-0.1,-0.1,-0.1,-0.1,-0.1,-0.1,0,-0.1}, {-0.1,-0.1,-0.1,-0.1,-0.1,-0.1}
,-0.1,1.0,-0.1},{-0.1,-0.1,-0.1,-1.0,-0.1,-0.1,-0.1,1.0,-0.1},{-0.1,-0.1,-0.1,-1.0,-0
.1, -0.1, -0.1, 1.0, -0.1, \{-0.1, -0.1, -0.1, -1.0, -0.1, -0.1, -0.1, 1.0, -0.1, \{-0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0
0.1, -0.1, -1.0, -0.1, -0.1, -0.1, 1.0, -0.1\}, \{\{-0.1, -0.1, -0.1, -1.0, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.1, -0.
, \{-0.1, -0.1, -0.1, -1.0, -0.1, -0.1, -0.1, 1.0, -0.1\}, \{-0.1, -0.1, -0.1, -1.0, -0.1, -0.1, -0.1, 1.0, -0.1\}
0.1, -0.1, -1.0, -0.1, -0.1, -0.1, 1.0, -0.1\}
double gamma = 0.999;
```

3/3/2016 Page Problem

Your goal is to develop a potential function Φ defined on each state so that shaping the reward function using Φ will cause PI to converge in as few iterations as possible starting from a 0-initialized value function. Your goal should be to make it converge in *one* PI iteration.