CS 397 Homework # 3
Due: March 15, by 5pm

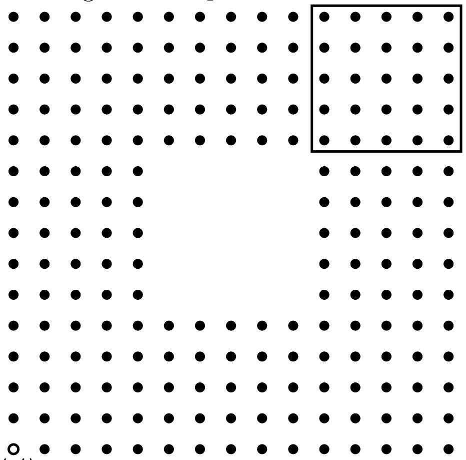
In this assignment, you are asked to study the multiple-stage decision making problem through the implementation of some software. You may use any language and playform that you prefer. Accounts and quotas should be available in the CS department instructional labs for all students registered in the class. This is an optional resource. When you are finished, you may simply e-mail me any output that is requested below. Also attach any source code that you wrote (no object code!).

The state space is a subset of a 2D integer grid. Let an integer, W, denote the *width* of the grid. To make things easier in the definitions and the software, assume that W is divisible by 3.

Suppose that the state space X is the set of all (i, j), such that

- i and j are both integers such that $1 \leq i, j \leq W$.
- at least one of the four inequalities must holds: $i \leq W/3$, $i > 2W/3$, $j \leq W/3$, and $j > 2W/3$.

These conditions yield an integer grid in which the middle 1/9 of the points are missing. For example, if $W = 15$, the set of states should be arranged in the plane as follows:



(1,1)

For the initial state, $x_I = (1, 1)$ (indicated by a white circle). Let the goal X_G be the set of all states, (i, j), for which $i > 2W/3$ and $j > 2W/3$ (in the example, they are the states inside the square).

Let $A = \{0, 1, 2, 3, 4\}$ be a set of actions, which denote:

0 Stay in the same location

1 Move right one unit

2 Move up one unit

3 Move left one unit

4 Move down one unit

Let $U = A \cup \{-1\}$, in which -1 represents the termination action. For each $x \in X$, let $U(x)$ contain -1, plus whichever actions are applicable from x (some will not be applicable due to boundaries).

For variations that involve nature, let $\Theta(x, u)$ represent the set of all actions in A that are applicable after performing the move implied by u. For example, if $x = (2, 2)$, and $u = 3$, then the decision maker is attempting to move to $(1, 2)$. From this state, there are three neighboring states, each of which corresponds to an action of nature. Thus, $\Theta(x, u)$ in this case is $\{0, 1, 2, 4\}$. The action $\theta = 3$ does not appear because there is no state to the left of $(1, 2)$.

The state transition equation, f, is formed by adding the effect of both u_k and θ_k. For example, if $x_k = (i, j)$, $u_k = 1$ and $\theta_k = 2$, then the x_{k+1} will be $(i + 1, j + 1)$. If θ_k had been 3, then the two actions would cancel, and x_{k+1} would be (i, j). For problems that do not involve nature, one can assume that $\theta_k = 0$, which means that nature does not interfere with the outcome. Note that the state never changes once u_T is applied, regardless of nature's actions.

For the loss functional, let $l(x_k, u_k) = 1$ unless $u_k = u_T$ (in this case, $l(x_k, u_T) = 0$). For the final stage, let $l_F(x_F) = 0$ if $x_F \in X_G$, otherwise let $l_F(x_F) = \infty$.

Assume that K is not given.

1. Assume that there are no nature effects and that $W = 15$. Use dynamic programming to compute an optimal action sequence. Give both the action sequence and the final resulting loss. How many dynamic programming iterations were required? (Think about how this compares with Dijkstra's algorithm.)

2. Assume that there is nondeterministic uncertainty and $W = 15$. Explain what happens in this case.

3. Assume that there is probabilistic uncertainty and $W = 15$. Assume that $P(\theta_k = 0) = 1/2$, regardless of u_k and x_k (except in the case of $u_k = u_T$, for which the state never changes). Assume that the remaining probability mass is distributed uniformly over the remaining actions in $\Theta(x_k, u_k)$. Compute the optimal (in terms of expected loss) strategy to within some reasonable accuracy. Show the optimal cost-to-values and the decision maker actions given by the optimal strategy for each state (this can be expressed as a 15x15 array with the center portion crossed out).

4. Conduct a simulation of the execution of the optimal strategy for the previous scenario. Generate at least 100 sample paths by making nature decisions based on the outcome of pseudorandom number outcomes. For each sample path, compute the loss received. Make a histogram with loss values along the x-axis and the frequency of their occurrence along the y-axis. For the histogram, you may either show a plot or simply list the frequencies. What is the average loss received? How does this compare with your computed expected loss for this strategy?

5. Have some fun with the code. How large can you make W and get it to finish? What happens if $P(\theta_k = 0)$ is chosen to be very small (assuming the other probabilities are adjusted accordingly)? If $P(\theta_k = 0)$ is close to one, do you get a result similar to that from Part 1?