BAMS 517: Assignment 3

(Each team submits one Word or PDF report and two R files on Canvas before the deadline. The Word/PDF report should include your answers to the questions with succinct explanations. Include the full names of the team members on the front page of the report. Name your R files as "Lastname_Firstname_HW3_Q1.R" and "Lastname_Firstname_HW3_Q2.R"; the name of any team member can be used.)

1. **(Machine Maintenance)** Suppose we have a machine that is either running or is broken down. If it runs throughout one week, it makes a gross profit of $100. If it fails during the week, gross profit is zeros.

   If it is running at the start of the week and we perform preventive maintenance, the probability that it will fail during the week is 0.4. If we do not perform such maintenance, the probability of failure is 0.65. However, maintenance will cost $30.

   When the machine is broken down at the start of the week, we have two options: (*i*) "repair" the machine immediately at a cost of $60, after which the machine can still fail during the week with a probability of 0.4; (*ii*) "replace" the machine immediately at a cost of $110 with a new machine, which is guaranteed to run through its first week of operation.

   Assume the machine is running at the start of the first week. Our goal is to find an optimal repair, replacement, and maintenance policy that maximizes the total profit over N weeks.

   (a) Formulate this problem as a finite-horizon MDP. Define the states, actions, rewards, and transition probabilities. Provide and briefly explain the Bellman optimality equations. (5 pts)

   (b) Develop R code to find the optimal solution (including the value function and policy). Report and discuss your solution for $N = 10$. Will the optimal policy change when $N = 20$? (10 pts)

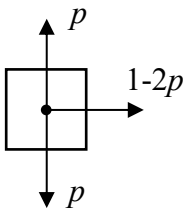
2. **(Robot Navigation)** A robot lives in a gridworld, which consists of 3*4 cells. For convenience, we call the cell in the *i*-th row and *j*-th column the cell (*i*,*j*). As illustrated in the following figure, there are three special cells. The "grey" cell represents a wall (column) that blocks the robot's path. If the robot enters the "green" cell, it receives a reward of $100. If the robot enters the "orange" cell, it incurs a penalty of $100 (a reward of -$100). The game ends when the robot reaches the green cell or the orange cell. There is a cost of $1 for staying in any other cell for one period.

-1	-1	-1	+100
-1		-1	-1
-1	-1	-1	-100

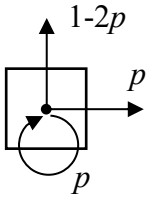
In other cells, the robot has four possible actions to take: Up (U), Down (D), Left (L), and Right (R). An action takes the robot to an adjacent cell. An action toward a wall (or border) is prevented. For example, in Cell (1,1), the feasible actions are {D, R}; in Cell (2,1), the feasible actions are {U, D}.

The robot's movement is inaccurate. Suppose the inaccuracy level is p. Then, the robot will move along the intended direction with probability $1 - 2p$ and along each perpendicular direction with probability p. If a perpendicular direction is invalid (blocked by a wall/border), the robot will stay in the same cell with the corresponding probability. Notice that the number of valid perpendicular directions varies with the cell and the action. Some examples are given below.

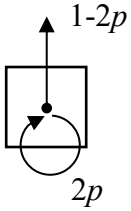
(i) Cell (2,3) and Action R: There are two valid perpendicular directions.



(ii) Cell (2,3) and Action U: There is only one valid perpendicular direction.



(iii) Cell (2,1) and Action U: There is no valid perpendicular direction.



Questions:

(a) Formulate this problem as an infinite-horizon MDP. Define the states, actions, and rewards. Write the transition probabilities for the action "Up" only (you may use a table or a set of formulas). Write down the Bellman optimality equations. (7 pts)

(b) Develop an R program and find the optimal solution using mdp_value_iteration(). Assume the following parameters: inaccuracy level $p = 0.02$ and discount factor $\delta = 0.99$. Report the optimal policy and optimal values-to-go. (10 pts)

(c) Solve the model in part (b) again under a new parameter: $p = 0.1$. Are there any changes in the optimal policy compared to part (b)? If so, describe the main differences and explain the main reason intuitively. (3 pts)

Hints:

1. This problem differs from the 2*2 Gridworld example discussed in class in multiple ways. In this problem, the robot does not always have four possible actions. For example, in cell (1,1), the only available actions are "Right" and "Down," while in the class example, the robot has four available actions in every cell. In a sense, the robot in our problem is more intelligent and can rule out obviously futile actions. (However, it can still make mistakes in its intended movement, which is captured by the inaccuracy level p.) To exclude an action from the feasible action set, you may set its reward to be a large negative number such as -9999 so that this action cannot be optimal.
2. Another difference is: in this problem, the process ends when the robot receives 100 or (-100) in the green (or orange) cell, while in the class example, the robot stays in the target cell and keeps collecting the reward 10 forever. In the standard infinite-horizon MDP model, the decision horizon never ends, which aligns with the class example but not our problem. There are different ways to handle this discrepancy. One is to make a terminal cell an absorbing state which traps the robot forever. This method requires setting the correct one-period reward in the absorbing state—if the robot receives $\$r$ for spending one period in that state, its total discounted reward for being in that state (forever) is $\$r/(1-\delta)$, where δ is the discount factor.
3. The function mdp_value_iteration() may introduce small numerical errors. For instance, instead of an optimal value-to-go 100, your model may return 99.95. Such small errors are allowed.