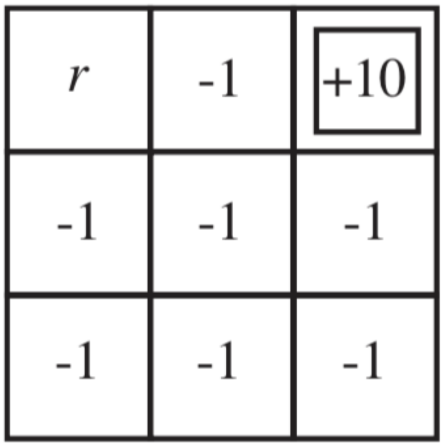
CSE 574
HW 3 Solutions
Due Friday September 28 at 5pm

Please submit on Blackboard. Show work for full credit. The programming assignment portion of the HW should be submitted as a separate file. See the class policies for late HW assignments as we will strictly enforce them.

Name:

ASU ID:

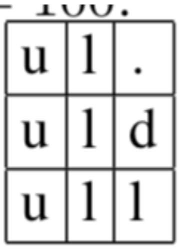
1. Markov decision processes:
Source: russell and norvig ex 17.8
Consider the 3 × 3 world shown in the following figure. The transition model is the following 80% of the time the agent goes in the direction it selects; the rest of the time it moves at right angles to the intended direction i.e. If it's intended direction is up it goes up 80% of the time up, 10% of the time goes to left and 10% of the time it goes to right.
Implement value iteration for this world for each value of r below. Use discounted rewards with a discount factor of 0.99. Show the policy obtained in each case. Explain intuitively why the value of r leads to each policy.
a. r=100
b. r = −3
c. r = 0
d. r = +3



Solution:

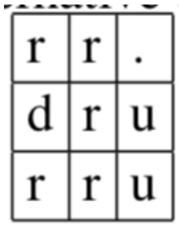
a. r=100

− 100.



u	l	.
u	l	d
u	l	l

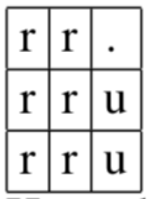
See the comments for part d. This should have been r = −100 to illustrate an alternative behavior:



r	r	.
d	r	u
r	r	u

Here, the agent tries to reach the goal quickly, subject to attempting to avoid the square (1, 3) as much as possible. Note that the agent will choose to move Down in square (1, 2) in order to actively avoid the possibility of "accidentally" moving into the square (1, 3) if it tried to move Right instead, since the penalty for moving into square (1, 3) is so great.

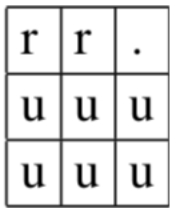
b. r = −3.



r	r	.
r	r	u
r	r	u

Here, the agent again tries to reach the goal as fast as possible while attempting to avoid the square (1, 3), but the penalty for square (1, 3) is not so great that the agent will try to actively avoid it at all costs. Thus, the agent will choose to move Right in square (1, 2) in order to try to get closer to the goal even if it occasionally will result in a transition to square (1, 3).

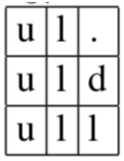
c. r = 0.



r	r	.
u	u	u
u	u	u

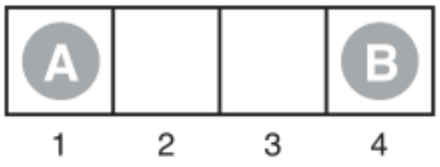
Here, the agent again tries to reach the goal as fast as possible, but will try to do so via a path that includes square (1, 3) if possible. This results from the fact that square(1, 3) does not incur the reward of −1 in all other non-goal states, so it reaching the goal via a path through that square can potentially have slightly greater reward than another path of equal length that does not pass through (1, 3).

d. r = 3.



u	l	.
u	l	d
u	l	l

2. **Value Iteration:** Recall the zero-sum, turn-taking games of question 6 in last homework. Let the players be A and B, and let R(s) be the reward for player A in state s. (The reward for B is always equal and opposite.) Let $U_A(s)$ be the utility of state s when it is A's turn to move in s, and let $U_B(s)$ be the utility of state s when it is B's turn to move in s. All rewards and utilities are calculated from A's point of view (just as in a minimax game tree).



a. Write down Bellman equations defining U_A (s) and U_B (s).
b. Explain how to do two-player value iteration with these equations, and define a suitable termination criterion.
c. Consider the game described in Figure 5.17 on page 197. Draw the state space (rather than the game tree), showing the moves by A as solid lines and moves by B as dashed lines. Mark each state with R(s). You will find it helpful to arrange the states (sA , sB) on a two-dimensional grid, using sA and sB as "coordinates."
d. Now apply two-player value iteration to solve this game, and derive the optimal policy.

Solution:

a. For U_A we have

$$U_A(s) = R(s) + \max_a \sum_{s'} P(s'|a,s) U_B(s')$$

and for U_B we have

$$U_B(s) = R(s) + \min_a \sum_{s'} P(s'|a,s) U_A(s') .$$

b. To do value iteration, we simply turn each equation from part (a) into a Bellman update and apply them in alternation, applying each to all states simultaneously. The process terminates when the utility vector for one player is the same as the previous utility vector *for the same player* (i.e., two steps earlier). (Note that typically U_A and U_B are not the same in equilibrium.)

c. The state space is shown in Figure S17.2.

d. We mark the terminal state values in bold and initialize other values to 0. Value iteration proceeds as follows:

	(1,4)	(2,4)	(3,4)	(1,3)	(2,3)	(4,3)	(1,2)	(3,2)	(4,2)	(2,1)	(3,1)
U_A	0	0	0	0	0	**+1**	0	0	**+1**	**–1**	**–1**
U_B	0	0	0	0	–1	**+1**	0	–1	**+1**	**–1**	**–1**
U_A	0	0	0	–1	+1	**+1**	–1	+1	**+1**	**–1**	**–1**
U_B	–1	+1	+1	–1	–1	**+1**	–1	–1	**+1**	**–1**	**–1**
U_A	+1	+1	+1	–1	+1	**+1**	–1	+1	**+1**	**–1**	**–1**
U_B	–1	+1	+1	–1	–1	**+1**	–1	–1	**+1**	**–1**	**–1**

and the optimal policy for each player is as follows:

	(1,4)	(2,4)	(3,4)	(1,3)	(2,3)	(4,3)	(1,2)	(3,2)	(4,2)	(2,1)	(3,1)
π^*_A	(2,4)	(3,4)	(2,4)	(2,3)	(4,3)		(3,2)	(4,2)			
π^*_B	(1,3)	(2,3)	(3,2)	(1,2)	(2,1)		(1,3)	(3,1)			

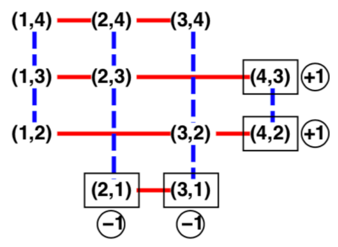


Figure S17.2 State-space graph for the game in Figure 5.17.

3. Finding Optimal Policies. Dynamic Programming and Optimal Control Vol 1 #1.3 on p 54

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 zero. 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 0.7. However, maintenance will cost \$20. When the machine is broken down at the start of the week, it may either be repaired at cost of \$40, in which case it will fail during the week with a probability of 0.4, or it may be replaced at a cost of \$150 by a new machine that is guaranteed to run through its first week of operation. Find the optimal repair, replacement, and maintenance policy that maximizes total profit over four weeks, assuming a new machine at the start of the first week.

Solution:

Problem 2 (Solution)

- States x: R : machine running, B : machine broken
- Control actions u:

$$\left.\begin{array}{rl} n: & \text{no maintenance} \\ m: & \text{maintenance} \end{array}\right\} \text{ if } x = R$$

$$\left.\begin{array}{rl} r: & \text{repair} \\ l: & \text{replace} \end{array}\right\} \text{ if } x = B$$

- Costs C:

$$\begin{array}{ccc} n & \to & 0 \\ m & \to & 20 \\ r & \to & 40 \\ l & \to & 150 \end{array}$$

- Gain ($\triangleq$ negative cost): -100 if not broken

Week 3

$x_3 = R$	$u_3 = n$	$C = 0.7(0) + 0.3(-100)$	$= -30$
	$u_3 = m$	$C = 20 + 0.6(-100)$	$= -40$
$x_3 = B$	$u_3 = r$	$C = 40 + 0.6(-100)$	$= -20$
	$u_3 = l$	$C = 150 + (-100)$	$=50$

$$\to \quad \begin{array}{ll} J_3(R) = -40 & \mu_3(R) = m \\ J_3(B) = -20 & \mu_3(B) = r \end{array}$$

Week 2

$x_2 = R$	$u_2 = n$	$C = 0 + 0.7(-20) + 0.3(-100 - 40)$	$= -56$
	$u_2 = m$	$C = 20 + 0.4(-20) + 0.6(-140)$	$= -72$
$x_2 = B$	$u_2 = r$	$C = 40 + 0.4(-20) + 0.6(-140)$	$= -52$
	$u_2 = l$	$C = 150 - 100 - 40$	$=10$

$$\rightarrow \quad J_2(R) = -72 \qquad \mu_2(R) = m$$
$$J_2(B) = -52 \qquad \mu_2(B) = r$$

<u>Week 1</u>

$x_1 = R$	$u_1 = n$	$C = 0 + 0.7(-52) + 0.3(-100 - 72)$	$= -88$
	$u_1 = m$	$C = 20 + 0.4(-52) + 0.6(-172)$	$= -104$
$x_1 = B$	$u_1 = r$	$C = 40 + 0.4(-52) + 0.6(-172)$	$= -84$
	$u_1 = l$	$C = 150 - 100 - 72$	$= -22$

$$\rightarrow \quad J_1(R) = -104 \qquad \mu_1(R) = m$$
$$J_1(B) = -84 \qquad \mu_1(B) = r$$

<u>Week 0</u>

$x_0 = R$ Machine is guaranteed to run in the 1^{st} week (since it is new)

$$\rightarrow \quad J_0(R) = -100 - 104 = -204 \quad , \quad \mu_0(R) = n$$

- Conclusion:
  - always maintain a running machine
  - always repair a broken machine
  - expected profit $\underline{-J_0(R) = 204}$

4. .

Let the state be the status of the first player. The state space, then, consists of the states

$$\begin{cases} x^1: & \text{player 1 having busted} \\ x^{1+i}: & \text{player 1 having stopped at sum } i \quad (i = 1, \ldots, 7) \\ x^{8+i}: & \text{current sum is } i\text{, but player 1 has not yet stopped} \quad (i = 1, \ldots, 6) \end{cases}$$

- The first player can roll 6 more times at most, after the initial throw. Before he rolls the dice he is in one of the above states. If in state x^8 or x^1, he has no choice. If in states x^2 to x^7, he has already stopped. If in states x^9 to x^{14}, he can apply a control (i.e., roll or stop), which will maximize $Pr\,(Win\,|\,x^8)$.
- For a given throw of the second player we can compute $Pr\,(Win\,|\,x^1)\,, \ldots, Pr\,(Win\,|\,x^8)$.
- Then going backwards in time (from the 6th roll) we calculate the strategy which maximizes $Pr(Win\,|\,x^i) \quad i = 14, \ldots, 9$. Let

$$P^*(Win\,|\,x^i) = \max_u Pr(Win\,|\,x^i)$$

Given that the initial throw of the second player is three:

$$\begin{aligned} Pr(Win\,|\,x^1) &= 0 & Pr(Win\,|\,x^5) &= 1/3 \\ Pr(Win\,|\,x^2) &= 1/3 & Pr(Win\,|\,x^6) &= 1/2 \\ Pr(Win\,|\,x^3) &= 1/3 & Pr(Win\,|\,x^7) &= 2/3 \\ Pr(Win\,|\,x^4) &= 1/3 & Pr(Win\,|\,x^8) &= 5/6 \end{aligned}$$

Let x_i be the state after the ith roll by player 1.

Stage 6

$$x_6 \in \{x^1, x^8\} \qquad \text{no controls can be applied}$$

Stage 5

$$x_5 \in \{x^1, x^7, x^8, x^{14}\} \qquad \text{control possible only for } x^{14}$$

$$Pr\{Win \,|\, x^{14}, u : stop\} = Pr\{Win \,|\, x^7\} = 2/3$$
$$Pr\{Win \,|\, x^{14}, u : roll\} = \frac{Pr\{Win \,|\, x^8\}}{6} = 5/36$$

(Note that $\mu_j(x^k)$ is independent of j.) We have

$$P^*(Win \,|\, x^{14}) = 2/3, \qquad \mu(x^{14}) : stop.$$

Stage 4

$$x_4 \in \{x^1, x^6, x^7, x^8, x^{13}, x^{14}\}$$

$$Pr\{Win \,|\, x^{13}, u : stop\} = Pr\{Win \,|\, x^6\} = 1/2$$
$$Pr\{Win \,|\, x^{13}, u : roll\} = \frac{Pr\{Win \,|\, x^8\} + P^*\{Win \,|\, x^{14}\}}{6} < 1/2$$

We have

$$P^*(Win \,|\, x^{13}) = \frac{1}{2}, \qquad \mu(x^{13}) : stop.$$

Stage 3

$$x_3 \in \{x^1, x^5, x^6, x^7, x^8, x^{12}, x^{13}, x^{14}\}$$

$$Pr\{Win \,|\, x^{12}, u : stop\} = Pr\{Win \,|\, x^5\} = 1/3$$
$$Pr\{Win \,|\, x^{12}, u : roll\} = \frac{Pr\{Win \,|\, x^8\} + P^*\{Win \,|\, x^{13}\} + P^*\{Win \,|\, x^{14}\}}{6} = 1/3$$

We have

$$P^*(Win \,|\, x^{12}) = 1/3, \qquad \mu(x^{13}) : stop \text{ or } roll.$$

Stage 3

$$x_3 \in \{x^1, x^4, x^5, x^6, x^7, x^8, x^{11}, x^{12}, x^{13}, x^{14}\}$$

$$Pr\{Win \,|\, x^{11}, u : stop\} = 1/3$$
$$Pr\{Win \,|\, x^{11}, u : roll\} = 7/18$$

We have

$$P^*(Win \,|\, x^{11}) = 7/18, \qquad \mu(x^{11}) : roll.$$

Finally,

$$P^*\{Win \,|\, x^{10}\} = 49/108, \quad \mu(x^{10}) : roll,$$
$$P^*\{Win \,|\, x^9\} = 343/648, \quad \mu(x^9) : roll.$$

Thus the optimal strategy is to throw until the sum is four or larger. If the sum is exactly four we can either roll or stop.

5.

Let the state x_k at stage k be the collection of four sets $\{C, T_0, T_1, T_2\}$ where

$C =$ the set of all coins which may still be counterfeit

$T_i =$ the set of all coins which have been tested i times

Although, given a state x_k, there are a huge number of potential controls, we choose to limit ourselves to controls which will yield potential information regarding the counterfeit coin. Letting c_k be the cardinality of set C_k, we can summarize the control constraints as

c_k	$U(c_k)$	Control Definition
6	$22, 11$	22: Test 2 coins from $T_{0,k}$ with 2 others from $T_{0,k}$ 11: Test 1 coin from $T_{0,k}$ with another from $T_{0,k}$
4	$\bar{1}\bar{1}, 2\bar{2}, 11$	$\bar{1}\bar{1}$: Test 1 coin from $T_{1,k}$ with another coin from $T_{1,k}$ $2\bar{2}$: Test 2 coins from $T_{0,k}$ with 2 from $T_{1,k}$
2	$1\bar{1}, 1\bar{\bar{1}}, \bar{1}\bar{\bar{1}}$	$1\bar{1}$: Test 1 coin from $T_{0,k}$ with 1 from $T_{1,k}$ $1\bar{\bar{1}}$: Test 1 coin from $T_{0,k}$ with 1 from $T_{2,k}$ $\bar{1}\bar{\bar{1}}$: Test 1 coin from $T_{1,k}$ with 1 from $T_{2,k}$

Note that odd cardinalities c_k cannot occur.

The disturbances in this problem are the actual results of the tests. Let the disturbance be w_k, where

$$w_k = \begin{cases} 1, & \text{if the test at stage } k \text{ yields an equal weight} \\ 0, & \text{if the test at stage } k \text{ yields an unequal weight} \end{cases}$$

The distribution of w_k depends on c_k and u_k.

To determine the optimal control strategy, given the control constraints, we see that

$$E\{\#\text{ of tests } | u_1 = 22\} = 1/3 \times 2 + 2/3 \times 3 = 8/3$$
$$E\{\#\text{ of tests } | u_1 = 11\} = 1/3 \times 2 + 2/3 \times 3 = 8/3$$

Thus, any u_1 control satisfying the $U(6)$ constraint will yield an equally acceptable solution.

6.

The distribution of w_k depends only on x_k and u_k, and the optimal value of the cost functional can be written in the form

$$\begin{aligned} J^*(x_0) &= \min_{\mu_0,\ldots,\mu_{N-1}} \underset{\substack{w_k \\ k=0,1,\ldots,N-1}}{E} \left\{ g_N(x_N) \prod_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right\} \\ &= \min_{\mu_0,\ldots,\mu_{N-1}} \Bigg[\underset{w_0}{E} \Big\{ g_0(x_0, \mu_0(x_0), w_0) \underset{w_1}{E} \big\{ g_1(x_1, \mu_1(x_1), w_1) \cdots \\ &\qquad \cdots \underset{w_{N-1}}{E} \{ g_{N-1}(x_{N-1}, \mu_{N-1}(x_{N-1}), w_{N-1})\; g_N(x_N) \} \cdots \big\} \Big\} \Bigg] \end{aligned}$$

Since the cost functions, g_k are non-negative, we can write

$$J^*(x_0) = \min_{\mu_0}\left[\underset{w_0}{E}\left\{g_0(x_0,\mu_0(x_0),w_0)\min_{\mu_1}\left[\underset{w_1}{E}\left\{g_1(x_1,\mu_1(x_1),w_1)\cdots\right.\right.\right.\right.$$
$$\left.\left.\left.\left.\cdots\min_{\mu_{N-1}}\left[\underset{w_{N-1}}{E}\{g_{N-1}(x_{N-1},\mu_{N-1}(x_{N-1}),w_{N-1})\ g_N(x_N)\}\right]\cdots\right\}\right]\right\}\right]$$

The optimal value $J^*(x_0)$ can be computed with the following DP algorithm:

$$J_N(x_N) = g_N(x_N)$$
$$J_k(x_k) = \min_{u_k}\underset{w_k}{E}\left\{g_k(x_k,u_k,w_k)\ J_{k+1}\left[f_k(x_k,u_k,w_k)\right]\right\}$$

where u_k obeys the control constraints. Note that we cannot simply minimize the log of the expected cost since $\log E(xy) \neq E[\log(xy)]$.

7. Proving Optimality of a Policy: Use proof by induction

1.21

A farmer annually producing x_k units of a certain crop stores $(1-u_k)x_k$ units of his production, where $0 \le u_k \le 1$, and invests the remaining u_kx_k units, thus increasing the next year's production to a level x_{k+1} given by

$$x_{k+1} = x_k + w_ku_kx_k, \qquad k = 0,1,\ldots,N-1.$$

The scalars w_k are independent random variables with identical probability distributions that do not depend either on x_k or u_k. Furthermore, $E\{w_k\} = \overline{w} > 0$. The problem is to find the optimal investment policy that maximizes the total expected product stored over N years

$$\underset{\substack{w_k\\k=0,1,\ldots,N-1}}{E}\left\{x_N + \sum_{k=0}^{N-1}(1-u_k)x_k\right\}.$$

Show the optimality of the following policy that consists of constant functions:

(a) If $\overline{w} > 1$, $\mu_0^*(x_0) = \cdots = \mu_{N-1}^*(x_{N-1}) = 1$.

(b) If $0 < \overline{w} < 1/N$, $\mu_0^*(x_0) = \cdots = \mu_{N-1}^*(x_{N-1}) = 0$.

(c) If $1/N \le \overline{w} \le 1$,

$$\mu_0^*(x_0) = \cdots = \mu_{N-\overline{k}-1}(x_{N-\overline{k}-1}) = 1,$$

$$\mu_{N-\overline{k}}^*(x_{N-\overline{k}}) = \cdots = \mu_{N-1}^*(x_{N-1}) = 0,$$

where $\overline{k}$ is such that $1/(\overline{k}+1) < \overline{w} \le 1/\overline{k}$.

Solution:

1.21

The DP algorithm is

$$J_N(x_N) = x_N$$

$$J_k(x_k) = k\left\{g_k(x_k, \mu_k(x_k), w_k) + \underset{w_k}{E}\left\{J_{k+1}\left[k\right]\right\}\right\}$$

Case 1: $\bar{w} > 1$. We claim that

$$\text{Claim:} \qquad J_{N-k}(x_{N-k}) = x_{N-k}(1+\bar{w})^k, \quad k = 1, \dots, N$$
$$0 = \cdots = N-1 = 1$$

The proof follows by induction.

$$\begin{aligned} J_{N-1}(x_{N-1}) &= N-1\left\{g_{N-1}(x_{N-1}, \mu_{N-1}(x_{N-1}), w_{N-1}) + \underset{w_{N-1}}{E}\{N-1\}\right\} \\ &= x_{N-1}\, N-1\{2 + (\bar{w}-1)u_{N-1}\} \\ &= x_{N-1}(1+\bar{w}), \qquad \textit{where } N-1 = 1. \end{aligned}$$

Assume that $J_{N-k}(x_{N-k}) = x_{N-k}(1+\bar{w})^k$. Then

$$\begin{aligned} J_{N-k-1}(x_{N-k-1}) &= N-k-1\left\{g_{N-k-1}(x_{N-k-1}, \mu_{N-k-1}(x_{N-k-1}), w_{N-k-1}) + (1+\bar{w}u_{N-k-1})(1+\bar{w})^k x_{N-k-1}\right\} \\ &= x_{N-k-1}N-k-1\left\{1 + (1+\bar{w})^k + \left[(1+\bar{w})^k\bar{w} - 1\right]u_{N-k-1}\right\} \\ &= x_{N-k-1}(1+\bar{w})^{k+1}, \qquad \textit{where } N-k-1 = 1. \end{aligned}$$

Case 2: $0 < \bar{w} < \frac{1}{N}$

$$\text{Claim:} \qquad J_{N-k}(x_{N-k}) = (k+1)x_{N-k}, \quad k = 1, \dots, N$$
$$0 = \cdots = N-1 = 0$$

The proof follows by induction. We have

$$J_{N-1}(x_{N-1}) = x_{N-1}\, N-1\{2 + (\bar{w}-1)u_{N-1}\} = 2x_{N-1}$$
$$\textit{where } N-1 = 0$$

Assume that $J_{N-k}(x_{N-k}) = (k+1)x_{N-k}$. Then

$$\begin{aligned} J_{N-k-1}(x_{N-k-1}) &= N-k-1\{g_{N-k-1}(x_{N-k-1}, \mu_{N-k-1}(x_{N-k-1}), w_{N-k-1}) + (k+1)(1+\bar{w}u_{N-k-1}x_{N-k-1})\} \\ &= x_{N-k-1}\, N-k-1\{(k+2) + [(k+1)\bar{w} - 1]u_{N-k-1}\} \\ &= (k+2)x_{N-k-1} \qquad \textit{where } N-k-1 = 0 \end{aligned}$$

Case 3: $\frac{1}{N} \leq \bar{w} \leq 1$
Apply the DP algorithm beginning with stage N. Proceed as in Case 2, setting the control equal to zero until

$$J_{N-\bar{k}-1}(x_{N-\bar{k}-1}) = x_{N-\bar{k}-1}\, N-\bar{k}-1\{(\bar{k}+2) + [(\bar{k}+1)\bar{w} - 1]u_{N-\bar{k}-1}\}$$

where $N - \bar{k} - 1$ is the first stage where $\bar{w} > 1/(\bar{k}+1)$. Since $(\bar{k}+1)\bar{w} - 1 > 0$, take

$$N - \bar{k} - 1 = 1$$
$$J_{N-\bar{k}-1}(x_{N-\bar{k}-1}) = (\bar{k}+1)(1+\bar{w})x_{N-\bar{k}-1}.$$

From this point, proceed as in Case 1. At each iteration the power of $(1+\bar{w})$ will be raised and the control will be set to one.