APP MATH 3020 Stochastic Decision Theory
Assignment 3

Due: Monday, 17 September, 2018, 10 a.m. $\boxed{\text{Total marks: 41}}$

Question 1 $\boxed{\text{2 marks}}$

Make sure that in all your answers you

$\boxed{1/2}$ (a) use full and complete sentences.

$\boxed{1/2}$ (b) include units where necessary.

$\boxed{1/2}$ (c) use logical arguments in your answers and proofs.

$\boxed{1/2}$ (d) structure your answers and assignment clearly and precisely.

Question 2 $\boxed{\text{19 marks}}$

Consider the problem:

$$
\begin{array}{lllllllll}
\min & z & = & 3x_1 & + & 2x_2 & + & \mathbb{E}_{\boldsymbol{\xi}}[15y_1 + 18y_2] & \\
\text{s.t.} & & & 3y_1 & + & 2y_2 & \leq & x_1 & \\
 & & & 2y_1 & + & 5y_2 & \leq & x_2 & \\
 & & & 0.6d_1 & \leq & y_1 & \leq & d_1 & \\
 & & & 0.8d_2 & \leq & y_2 & \leq & d_2 & \\
 & & & & & x_i & \geq & 0 & \text{for } i = 1,2 \\
 & & & & & y_i & \geq & 0 & \text{for } i = 1,2,
\end{array}
$$

where the random vector $\boldsymbol{\xi} = (d_1, d_2)^\top$ has two realisations:

$$
\begin{aligned}
\boldsymbol{\varepsilon}_1 = (d_1, d_2) = (4,4)^\top \quad & \text{with probability } 0.5, \\
\boldsymbol{\varepsilon}_2 = (d_1, d_2) = (6,8)^\top \quad & \text{with probability } 0.5.
\end{aligned}
$$

$\boxed{19}$ (a) Perform Steps 0, 1, and 2 of the L-shaped algorithm on the above problem. For the purpose of this assignment, do each step only once; that is, stop when the algorithm asks you to either go back to Step 1 or proceed to Step 3. Explain all your working and provide all MATLAB code and relevant output.

Solution: We write the first stage LP as

$$
\begin{array}{llllllll}
\min & z & = & 3x_1 & + & 2x_2 & + & \theta \\
\text{s.t.} & & & & & x_i & \geq & 0, \quad \text{for } i = 1,2.
\end{array}
$$

where initially we set $\theta = 0$.

Step 0. [1 mark] We initialise by setting

$$
t = s = \nu = 0, \quad \boldsymbol{D}_0 = \boldsymbol{0}, \quad d_0 = 0, \quad \boldsymbol{E}_0 = \boldsymbol{0}, \quad e_0 = 0.
$$

[1 mark for all five eqns being there, 0 if at least one is missing]

Step 1. [**4 marks**] We set $\nu = \nu + 1 = 1$ [1] and solve

$$\begin{array}{rrcl} \min & z & = & 3x_1 + 2x_2 \\ \text{s.t.} & & & x_i \geq 0, \quad \text{for } i = 1, 2, \end{array}$$

to get $\boldsymbol{x}^{(1)} = (0,0)^\top$ with $z^{(1)} = 0$ [1 for LP, 1 for soln/val] . This optimal solution can be obtained simply by inspection.

As there are no optimality cuts yet, we set $\theta^{(1)} = -\infty$ [1] .

Step 2. [**10 marks, not including code/output**]

There are two realisations, so we will need to solve two LPs, to determine the feasibility of the second-program.

First realisation. [**6 marks**]

For $\boldsymbol{\varepsilon}_1 = (4,4)^\top$ and $i = 1, \ldots, 6$, we solve [2 marks for LP]

$$\left.\begin{array}{lrl} \min\ w & = & \sum_{i=1}^{6} v_i \\ \text{s.t.} & & -v_1 + 3y_1 + 2y_2 \leq x_1 = 0 \\ & & -v_2 + 2y_1 + 5y_2 \leq x_2 = 0 \\ & & v_3 + y_1 \geq 2.4 \\ & & v_4 + y_2 \geq 3.2 \\ & & -v_5 + y_1 \leq 4 \\ & & -v_6 + y_2 \leq 4 \\ & & \text{for } i = 1, \ldots 6,\ v_i \geq 0 \\ & & y_1, y_2 \geq 0. \end{array}\right\} \qquad (1)$$

Note that here we have either subtracted $v_i \geq 0$ or added $v_i \geq 0$ in order to satisfy the constraints of the second-stage program. By MATLAB, the solution of the LP (1) is $\boldsymbol{v} = (0, 0, 2.4, 3.2, 0, 0)^\top$ and $w_1^{(1)} = 5.6$. [1 mark for primal solution]

As $w_1^{(1)} > 0$, this indicates that given $\boldsymbol{\varepsilon}_1$ and the first-stage solution $\boldsymbol{x}^{(1)}$, there is no feasible solution $\boldsymbol{y} \geq \boldsymbol{0}$. The dual LP has solution $\boldsymbol{\sigma}_1^{(1)} = (-0.6993, -0.7264, -1, -1, 0, 0)^\top$. [1 for dual soln; please be careful because students might order variables differently.]

Also, note that here the first two entries of $\boldsymbol{\sigma}_1^{(1)}$, highlighted in red, do not matter at all; that is, they could take any values. This is because the objective function of the dual is in the form $(\boldsymbol{h}_1 - T(\boldsymbol{\varepsilon}_1)\boldsymbol{x}^{(1)})^\top \boldsymbol{\sigma}_1^{(1)} = \boldsymbol{h}_1^\top \boldsymbol{\sigma}_1^{(1)}$.

Here, we have

$$\boldsymbol{h}_1 = (0, 0, -2.4, -3.2, 4, 4)^\top, \quad T(\boldsymbol{\varepsilon}_1) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$\boldsymbol{D}_1 = (\boldsymbol{\sigma}_1^{(1)})^\top T(\boldsymbol{\varepsilon}_1) = (0.6993, 0.7264)$, [This might change depending on which $\boldsymbol{\sigma}_1^{(1)}$ we take!]

$d_1 = (\boldsymbol{\sigma}_1^{(1)})^\top \boldsymbol{h}(\boldsymbol{\varepsilon}_1) = 5.6.$ [1 mark for $\boldsymbol{D}_1$ and d_1]

We set $t = t + 1 = 1$ and create the feasibility cut

$$\boldsymbol{D}_1\boldsymbol{x} = 0.6993x_1 + 0.7264x_2 \geq 5.6 = d_1. \text{ [1]} \tag{2}$$

Second realisation. [4 marks]

For $\boldsymbol{\varepsilon}_2 = (6, 8)^\top$ and $i = 1, \ldots, 6$, we solve

$$\left.\begin{array}{rcrcrcrcl}
\min\ w & = & \sum_{i=1}^{6} v_i & & & & & & \\
\text{s.t.} & & - & v_1 & + & 3y_1 & + & 2y_2 & \leq & x_1 = 0 \\
 & & - & v_2 & + & 2y_1 & + & 5y_2 & \leq & x_2 = 0 \\
 & & & v_3 & + & y_1 & & & \geq & 3.6 \\
 & & & v_4 & & & + & y_2 & \geq & 6.4 \\
 & & - & v_5 & + & y_1 & & & \leq & 6 \\
 & & - & v_6 & & & + & y_2 & \leq & 8 \\
 & & & & & \text{for } i = 1, \ldots 6, & & v_i & \geq & 0 \\
 & & & & & & & y_1, y_2 & \geq & 0.
\end{array}\right\} \tag{3}$$

The solution of the LP (3) is $\boldsymbol{v} = (0, 0, 3.6, 6.4, 0, 0)^\top$, and $w_2^{(1)} = 10$. [1 for primal soln]

As $w_2^{(1)} > 0$, this indicates that, given $\boldsymbol{\varepsilon}_2$, there is no feasible solution $\boldsymbol{y} \geq \boldsymbol{0}$ for the first-stage solution $\boldsymbol{x}^{(1)}$. The dual LP has solution $\boldsymbol{\sigma}_2^{(1)} = (-0.6421, -0.6208, -1, -1, 0, 0)$. [1 mark for dual solution]

The same story applies here: note that here the first two entries of $\boldsymbol{\sigma}_2^{(1)}$, highlighted in red, do not matter at all; that is, they could take any values. This is because the objective function of the dual is in the form $(\boldsymbol{h}_2 - T(\boldsymbol{\varepsilon}_2)\boldsymbol{x}^{(1)})^\top\boldsymbol{\sigma}_2^{(1)} = \boldsymbol{h}_2^\top\boldsymbol{\sigma}_2^{(1)}$.

Here, we have

$$\boldsymbol{h}_2 = (0, 0, -3.6, -6.4, 6, 8)^\top, \quad T(\boldsymbol{\varepsilon_2}) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$\boldsymbol{D}_2 = \boldsymbol{\sigma}_2^{(1)} T(\boldsymbol{\varepsilon}_2) = (0.6421, 0.6208)$, [This might change depending on which $\boldsymbol{\sigma}_2^{(1)}$ we take!]

$d_2 = \boldsymbol{\sigma}_2^{(1)} \boldsymbol{h}(\boldsymbol{\varepsilon}_2) = 10.$

We set $t = t + 1 = 2$, and create the feasibility cut [1 mark for feasibility cut]

$$\boldsymbol{D}_2\boldsymbol{x} = 0.6421x_1 + 0.6208x_2 \geq 10 = d_2, \tag{4}$$

and return to **Step 1.** [1 mark for returning to **Step 1**]

MATLAB code and output [4 marks, 0.5 for each of eight parts]

Step 2, for the first realisation ε_1:

(a) The MATLAB code to solve the LP (1) is

```
f = [1, 1, 1, 1, 1, 1, 0, 0];
A = [-1  0  0  0  0  0  3  2;
      0 -1  0  0  0  0  2  5;
      0  0 -1  0  0  0 -1  0;
      0  0  0 -1  0  0  0 -1;
      0  0  0  0 -1  0  1  0;
      0  0  0  0  0 -1  0  1];
b = [0; 0; -2.4; -3.2; 4; 4];
LB = [0; 0; 0; 0; 0; 0];
[x, val] = linprog(f, A, b, [], [], LB, [])
```

(b) The associated MATLAB output is

```
x =
         0
         0
    2.4000
    3.2000
         0
         0
         0
         0
val =
    5.6000
```

(c) The MATLAB code for solving the dual of the LP (1) is

```
f_dual = -b';
A_dual = A';
b_dual = f';
UB_dual = [0; 0; 0; 0; 0; 0];
[x_dual, val_dual] = linprog(f_dual, A_dual, b_dual, [], [], [], UB_dual)
```

(d) The associated MATLAB output is

```
x_dual =
   -0.6993
   -0.7264
   -1.0000
   -1.0000
   -0.0000
    0.0000

val_dual =
   -6.4000
```

Step 2, for the second realisation ε_2:

(e) The MATLAB code to solve the LP (3) is

```
f = [1, 1, 1, 1, 1, 1, 0, 0];
A = [-1  0  0  0  0  0  3  2;
      0 -1  0  0  0  0  2  5;
      0  0 -1  0  0  0 -1  0;
      0  0  0 -1  0  0  0 -1;
      0  0  0  0 -1  0  1  0;
      0  0  0  0  0 -1  0  1];
b = [0; 0; -3.6; -6.4; 6; 8];
LB = [0; 0; 0; 0; 0; 0; 0; 0];
[x, val] = linprog(f, A, b, [], [], LB, [])
```

(f) The associated MATLAB output is

```
x =
    0.0000
    0.0000
    3.6000
    6.4000
    0.0000
    0.0000
    0.0000
    0.0000
val =
   10.0000
```

(g) The MATLAB code for solving the dual of the LP (3) is

```
f_dual = -b';
A_dual = A';
b_dual = f';
UB_dual = [0; 0; 0; 0; 0; 0];
[x_dual, val_dual] = linprog(f_dual, A_dual, b_dual, [], [], [], UB_dual)
```

(h) The associated MATLAB output is

```
x_dual =
   -0.6421
   -0.6208
   -1.0000
   -1.0000
    0.0000
    0.0000

val_dual =
  -10.0000
```

Question 3 $\boxed{\text{16 marks}}$

The Adelaide University Mathematics Society is having a BBQ; there are loads of free food. You stand in line to get a hot dog, and n students are in front of you. With probability p the student at the head of the queue will finish being served in the next minute, independently of what happens in all other minutes. At the beginning of every minute, you need to decide to continue waiting or leave.

Suppose there is a cost of \$$a$ for every minute spent waiting, and the hot dog is \$$b$. Let J_n denote the expected reward obtained by employing an optimal waiting policy when there are n people ahead of you in the queue.

$\boxed{5}$ (a) Write the optimality equation for $n = 0$ and $n > 0$, with justification.

Solution: For $n = 0$, we have $J_0 = \max\{0, b\} = b$, which is the maximum between getting 0 (if we leave immediately) and getting b if we stay in line and get the hot dog. [1 for equation, 1 for reasoning] .

For $n > 0$,

$$J_n = \max\{0, -a + pJ_{n-1} + (1-p)J_n\}. \text{ [1]}$$

The first argument on the right-hand side corresponds to the reward of zero if we leave. For the second, if we wait there is a cost of \$$a$ for that minute [1] ; furthermore, with probability p someone leaves and with probability $1-p$ no one does [1] .

$\boxed{3}$ (b) Show that the optimality equation can be rewritten as

$$J_n = \max\{J_{n-1} - a/p, 0\} \qquad \text{for } n > 0.$$

Solution: For $n > 0$, if $-a + pJ_{n-1} + (1-p)J_n < 0$ then $J_n = 0$. Note that the condition $-a + pJ_{n-1} + (1-p)J_n < 0$ implies $-a + pb < 0$ (setting $n = 1$). Thus, the above equation holds trivially. [1]

Now consider the other case, when $-a + pJ_{n-1} + (1-p)J_n \geq 0$. Then, using the optimality equation we have

$$\begin{aligned} & J_n = -a + pJ_{n-1} + (1-p)J_n \text{ [1]} \\ \Leftrightarrow \quad & 0 = -a/p + J_{n-1} - J_n \text{ [1]} \\ \Leftrightarrow \quad & J_n = J_{n-1} - a/p = \max\{J_{n-1} - a/p, 0\} \end{aligned}$$

as required.

$\boxed{2}$ (c) By induction, the above implies $J_n \leq J_{n-1}$. Explain why intuitively this is expected.

Solution: The cost is always increasing with n [1] , whereas the payoff (the value of our delicious hotdog) does not. [1]

$\boxed{6}$ (d) Determine the threshold N such that the form of the optimal policy is to wait only if $n \leq N$. Find J_n in terms of a, b and p.

Solution: As $J_0 = b > 0$ and J_n is a decreasing function of n, there exist some threshold N such that for $n < N$ we have $J_n > 0$ and for $n \geq N$ we have $J_n \leq 0$, where N could be ∞.

Writing J_n recursively, we have

$$\begin{aligned} J_n &= \max\{J_{n-1} - a/p, 0\} \\ &= \max\{J_{n-2} - 2a/p, 0\} \\ &= \cdots \\ &= \max\{J_0 - na/p, 0\} \\ &= \max\{b - na/p, 0\}. \ [2] \end{aligned}$$

Hence, we stop waiting the first time, N, that $b - Na/p \leq 0$, [1] which implies $N \geq bp/a$. [1] We have

$$J_n = \begin{cases} b - \dfrac{na}{p} & n < N, \ [1] \\ 0 & \text{otherwise. } [1] \end{cases}$$

Question 4 $\boxed{\text{4 marks}}$

The bad news is you have just been bitten by a poisonous spider. The good news is you have m potion bottles which might just save your life. If you drink the ith bottle, there is a probability of α_i that you will stay alive, a probability of β_i that you will die instantaneously, and a probability of $1 - \alpha_i - \beta_i$ that the potion will do absolutely nothing.

$\boxed{4}$ (a) Determine the order in which you should drink the potion bottles to maximise your probability of staying alive, in terms of the ratios α_i/β_i.

Solution: Consider drinking the bottles in some order

$$b_1, b_2, \ldots b_i, b_j, b_k, b_\ell, \ldots, b_m,$$

and then consider if instead, you adopted the order

$$d_1, d_2, \ldots d_i, d_k, d_j, d_\ell, \ldots, d_m,$$

where bottles b_j and b_k have swapped. The outcome up to and including bottle i is identical. From that point on, the respective probabilities of staying alive of the two orders are

$$\alpha_j + (1 - \alpha_j - \beta_j)\alpha_k + (1 - \alpha_j - \beta_j)(1 - \alpha_k - \beta_k)P_{\ell,\ldots,n}$$

and

$$\alpha_k + (1 - \alpha_k - \beta_k)\alpha_j + (1 - \alpha_k - \beta_k)(1 - \alpha_j - \beta_j)P_{\ell,\ldots,n},$$

where $P_{\ell,\ldots,m}$ denotes the probability of staying alive when drinking the remaining bottles in order $b_\ell, \ldots, d_m$. [2]

Hence, we would choose to drink bottle j before bottle k if

$$\alpha_j + (1 - \alpha_j - \beta_j)\alpha_k \geq \alpha_k + (1 - \alpha_k - \beta_k)\alpha_j, \text{ [1]}$$

which is equivalent to

$$\frac{\alpha_j}{\beta_j} \geq \frac{\alpha_k}{\beta_k}. \text{ [1]}$$

Note: As this is independent of where it occurred in the sequence, we would open doors in order of decreasing value of α_i/β_i. This makes sense intuitively, as α_i is the probability of the ith bottle saving you and β_i leading to an instant death!