**CSCI 5654-Fall16**: Assignment #5
**Robert Werthman**

---

**P1.**

**(A)**

Solve the initial linear equation and then branch on any variables that have a fractional value. Add the constrains of the fractional variables, solve the linear equation, and repeat until the solution is integral and not fractional.

| Branches | Solution | Optimal value |
|---|---|---|
| | $x = [-5 \quad -5 \quad 1.5 \quad -.5]$ | 2.5 |
| $x_4 \leq -1$ | $x = [-5 \quad -5 \quad 2 \quad -1]$ | 2 |
| $x_4 \geq 0$ | $x = [-2 \quad -4 \quad 0 \quad 0]$ | 2 |

The optimal value of the objective function is 2. One solution that leads to this value is:

$$x_1 = 1 \quad x_2 = 1 \quad x_3 = 1 \quad x_4 = 1$$

**(B)**

| Branches | Solution | Optimal value |
|---|---|---|
| | $x = [1.3333 \quad 1 \quad 1 \quad .6667]$ | 6.3333 |
| $x_4 \geq 1$ | $x = [1 \quad 1 \quad 1 \quad 1]$ | 6 |
| $x_4 \leq 0$ | $x = [0 \quad 1 \quad 1 \quad 0]$ | 3 |
| $x_1 \leq 1$ | $x = [1 \quad 1 \quad 1 \quad 1]$ | 6 |
| $x_1 \geq 2$ | Infeasible | |

The optimal value of the objective function is 6. One solution that leads to this value is:

$$x_1 = 1 \quad x_2 = 1 \quad x_3 = 1 \quad x_4 = 1$$

**P2.**

**Dictionary # 1**

First, we choose $x_1$ because it is a variable with a fractional solution. We rewrite the equation for $x_1$ as:

$$0.666667x_5 - 0.333333x_4 + x_1 = 0.666666666667$$

Next we rewrite the above equation in terms of an integer part and a fractional part.

$$(0x_5 - x_4 + x_1) + (0.666667x_5 + 0.777777x_4) = 0 + 0.666666666667$$

The fractional part $(0.666667x_5 + 0.777777x_4) \geq 0.666666666667$. The cutting plane is then given by the equation:

$$(0.666667x_5 + 0.777777x_4) + w_6 = 0.666666666667$$

**Dictionary # 2**

Equations for variables with fractional solutions:

$$\begin{aligned}
-0.333333x_8 - 0.666667x_9 + 0.333333x_3 + x_4 &= 4.33333333333 \\
0.333333x_8 - 0.333333x_9 + 2.666667x_3 + x_5 &= 8.66666666667 \\
0.333333x_8 + 0.666667x_9 - 0.333333x_3 + x_1 &= 5.66666666667 \\
-0.333333x_8 + 0.333333x_9 - 2.666667x_3 + x_2 &= 1.33333333333
\end{aligned}$$

Equations written with integral and fractional parts:

$$\begin{aligned}
(-x_8 - x_9 + 0x_3 + x_4) + (0.777777x_8 + 0.444443x_9 + 0.333333x_3) &= 4 + .33333333333 \\
(0x_8 - x_9 + 2x_3 + x_5) + (0.333333x_8 + 0.777777x_9 + 666667x_3) &= 8 + .66666666667 \\
(0x_8 + 0x_9 - x_3 + x_1) + (0.333333x_8 + 0.666667x_9 + 0.777777x_3) &= 5 + .66666666667 \\
(-x_8 + 0x_9 - 3x_3 + x_2) + (0.777777x_8 + 0.333333x_9 + 0.444443x_3) &= 1 + .33333333333
\end{aligned}$$

Cutting planes for the above equations:

$$\begin{aligned}
(0.777777x_8 + 0.444443x_9 + 0.333333x_3) + w_{10} &= .33333333333 \\
(0.333333x_8 + 0.777777x_9 + 666667x_3) + w_{11} &= .66666666667 \\
(0.333333x_8 + 0.666667x_9 + 0.777777x_3) + w_{12} &= .66666666667 \\
(0.777777x_8 + 0.333333x_9 + 0.444443x_3) + w_{13} &= .33333333333
\end{aligned}$$

**P3.**

**(A)**

Let $x_i$ be node $n_i$. If $x_i = 1$ then there is a hospital at that node. If $x_i = 0$ then there is no hospital at that node, but there should be at least one other node $x_j = 1$ and the distance to that node $W(i, j)$ should be between 0 and 1.

This is a 0-1 Integer Linear Program given by the following formulation:

$$\begin{array}{lll} \min & \sum_{j=1}^{n}(cost_j * node_j) & \\ \text{s.t.} & & \\ & \sum_{j=1}^{n} I(W(i,j) \leq 1) * n_j & \geq 1 \text{ for all } i = 1...n \\ & n_j & \in \{0, 1\} \end{array}$$

For the objective function, we want to minimize the cost of placing hospitals. The constraint

$$\sum_{j=1}^{n} I(W(i,j) \leq 1) * n_j \geq 1 \text{ for all } i = 1...n$$

says that for each node $i$ we want to make sure that there is a node $j$ that has a hospital $n_j = 1$ and is within 1 hour of it $I(W(i, j) \leq 1) = 1$.

**(B)**

The code to the solution is at the end of this document. The solution itself is the vector

$$x = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1]$$

This means in order to minimize the cost of building the hospitals within the driving time of 1 hour for each node, hospitals should be placed at node 6 and node 8. The total cost of the hospitals ends up being 2.1 million dollars.

**P4.**

**(A)** We add the indicator variable $z_1 ... z_n$ to the problem. If $x_i == 0$ then $z_i = 0$ else $z_i = 1$. In this problem we want to minimize the sum of $z_i$ and this occurs best when $x_i$ is 0.

$$\begin{aligned} \min \quad & \sum z \\ \text{s.t.} \quad & \\ & Ax \quad \leq b \\ & x \quad \leq u * z \\ & x \quad \geq \ell * z \\ & x \quad \leq u \\ & x \quad \geq \ell \\ & z \quad \in \{0, 1\} \end{aligned}$$

The additional constraints

$$x \leq u * z$$
$$x \geq \ell * z$$

say that if $x_i$ is negative then $z_i$ must equal 1, referring to $x \geq \ell * z$. If $x_i$ is positive then $z_i$ must be equal to 1, referring to $x \leq u * z$. But if x is 0 then z should be equal to 0 because this is a minimization problem.

**(B)**

We can use the same problem formulation from above except we make it a maximization problem instead of a minimization.

$$\begin{aligned} \max \quad & \sum z \\ \text{s.t.} \quad & \\ & Ax \quad \leq b \\ & x \quad \leq u * z \\ & x \quad \geq \ell * z \\ & x \quad \leq u \\ & x \quad \geq \ell \\ & z \quad \in \{0, 1\} \end{aligned}$$

The additional constraints

$$x \leq u * z$$
$$x \geq \ell * z$$

say that if x is negative then z must equal 1, referring to $x \geq \ell * z$. If x is positive then z must be equal to 1, referring to $x \leq u * z$. But if x is 0 then z should be equal to 1 because this is a maximization problem.

**(C)**

In order to not satisfy the inequality $a \leq x \leq b$, x needs to be either $\ell \leq x \leq a$ or $b \leq x \leq u$. The

mixed integer program would look like:

$$\begin{array}{ll} \max & c^t x \\ \text{s.t.} & \\ & Ax \leq b \\ & x \geq \ell * w + (1-w) * b \\ & x \leq a * w + (1-w) * u \\ & w \in \{1, 0\} \end{array}$$

The additional constraints

$$\begin{aligned} x &\geq \ell * w + (1-w) * b \\ x &\leq a * w + (1-w) * u \end{aligned}$$

say that if $w_i = 1$ then $\ell \leq x \leq a$. If $w_i = 0$ then $b \leq x \leq u$.

**P5.**

We have to find a subset $S$ that contains at least one element in the set $S_i$ for $i = 1, \ldots, k$ and the sum of the elements in $S$ is minimized.

$$\begin{array}{ll} \min & \sum_{i=1}^{n}(i * x_i) \\ \text{s.t.} & \\ & \sum_{i=1}^{n}(i * x_{ij}) \quad \geq 1 \text{ for all } j = 1...k \\ & x_i \qquad\qquad\quad \in \{0, 1\} \end{array}$$

$x_i = 1$ if the element $i$ is in the subset $S$ otherwise $x_i = 0$. The constraint

$$\sum_{j=1}^{n}(j * x_{ji}) \geq 1 \text{ for all } i = 1...k$$

says that for each set $S_i$, the subset $S$ must contain at least one element from $S_i$. Solving the above example in the $0 - 1$ ILP we get the solution vector

$$x = [1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

This means the subset $S = \{1, 2\}$ which sums to 3.

```
% Problem 1 Part a
options = optimoptions('linprog','Algorithm','dual-simplex');
% p1 a
f = [2 -3 -2 -1];
A = [1 -1 1 0
     1 -2 -1 1
     1 -1 -1 -1];
b = [5 3 -1];
lb = [-5 -5 -5 -5];
ub = [5 5 5 5];
[x, fval] = linprog(-f,A,b,[],[],lb,ub,options)

% p1 a x_4 <= -1
lb = [-5 -5 -5 -5];
ub = [5 5 5 -1];
[x, fval] = linprog(-f,A,b,[],[],lb,ub,options)

% p1 a x_4 >= 0
lb = [-5 -5 -5 0];
ub = [5 5 5 5];
[x, fval] = linprog(-f,A,b,[],[],lb,ub,options)
```

```
% Problem 1 Part b
options = optimoptions('linprog','Algorithm','dual-simplex');
f = [2 0 3 1];
A = [1 -1 0 1
     0 2 0 -1
     1 0 -1 -2
     -1 0 0 1];
b = [1 2 -1 1];
ub = [2 1 1 1];
lb = [-2 -1 0 -1];

[x, fval] = linprog(-f,A,b,[],[],lb,ub,options)

% x_4 <= 0
ub = [2 1 1 0];
lb = [-2 -1 0 -1];
```

```
17
18 [x, fval] = linprog(-f,A,b,[],[],lb,ub,options)
19
20 % x_4 >= 1
21 ub = [2 1 1 1];
22 lb = [-2 -1 0 1];
23
24 [x, fval] = linprog(-f,A,b,[],[],lb,ub,options)
25
26 % x_1 <= 1
27 ub = [1 1 1 1];
28 lb = [-2 -1 0 -1];
29 [x, fval] = linprog(-f,A,b,[],[],lb,ub,options)
30
31 % x_1 >= 2
32 ub = [2 1 1 1];
33 lb = [2 -1 0 -1];
34 [x, fval] = linprog(-f,A,b,[],[],lb,ub,options)
```

```
1 % Problem 3 Part b
2 f = [3 3 1.5 1 1.2 1.3 .9 .8];
3 A = [1 1 1 0 0 1 0 0
4      1 1 1 1 0 1 0 0
5      1 1 1 0 1 1 1 1
6      0 1 0 1 0 1 0 0
7      0 0 1 0 1 0 1 1
8      1 1 1 1 0 1 1 1
9      0 0 1 0 1 1 1 1
10     0 0 1 0 1 1 1 1];
11 A = A.*(-1);
12 b = [-1 -1 -1 -1 -1 -1 -1 -1]';
13 intcon = 8;
14 ub = [1 1 1 1 1 1 1 1];
15 lb = [0 0 0 0 0 0 0 0];
16
17 [x,fval] = intlinprog(f,intcon,A,b,[],[],lb,ub)
```

```
1 % Problem 5
2 f = [1 2 3 4 5 6 7 8 9 10];
3 A = [-1 0 -1 0 0 -1 0 0 0 0
4      0 -1 0 0 0 0 -1 -1 0 0
5      -1 0 0 0 0 0 0 -1 -1 0
6      -1 0 -1 0 -1 -1 0 0 0 0];
7 b = [-1 -1 -1 -1]';
8 ub = [1 1 1 1 1 1 1 1 1 1];
9 lb = [0 0 0 0 0 0 0 0 0 0];
10 intcon = 10;
11
12 [x,fval] = intlinprog(f,intcon,A,b,[],[],lb,ub)
```