Homework 1

Problem: Pick a small example from the AMPL book and write the corresponding LP in its original form, standard form and canonical form.

Solution: For this problem I used the very first LP presented in the book. It is presented as such:

$$\begin{aligned} \max \quad & 25X_B + 30X_c \\ \text{Subject To:} \quad & (1/200)X_B + (1/140)X_C \leq 40 \\ & 0 \leq X_B \leq 6000 \\ & 0 \leq X_C \leq 4000 \end{aligned}$$

Converting this to canonical form is easier than standard form so we'll start there.

Canonical Form

For this we want the following setup:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \end{aligned}$$

First we'll handle the conversion from max to min.

$$\max 25X_B + 30X_C = \min -25X_B - 30X_C$$

From here we need to account for something, we need all less than inequalities for our constraints however we have double inequalities. So we need to adjust those. Double inequalities aren't anything fancy really, they're just two sets of inequalities written in a more concise way.

On top of that, we need to capture all of the coefficients in these inequalities. Some of these constraints only have a single variable, but in a way they still contain both. The one that isn't present can be represented with a simple 0 coefficient. Capturing all of that information, let's begin.

First off,

$$\begin{aligned} 0 \leq X_B \leq 6000 &\iff 0 \leq X_B, X_B \leq 6000 \\ 0 \leq X_C \leq 4000 &\iff 0 \leq X_C, X_C \leq 4000 \end{aligned}$$

And,

$$\begin{aligned} 0 \leq X_B &\iff 0 \leq X_B + 0X_C \\ 0 \leq X_C &\iff 0 \leq X_C + 0X_B \end{aligned}$$

Now let's rewrite all of our constraints. I'll also be flipping these inequalities to ensure all of them are in the same direction.

$$\begin{aligned}
\frac{1}{200}X_B + \frac{1}{140}X_C &\leq 40 \\
X_B + 0X_C &\leq 6000 \\
0X_B + X_C &\leq 4000 \\
-X_B + 0X_C &\leq 0 \\
0X_B - X_C &\leq 0
\end{aligned}$$

Now we can rewrite all of what we have in vector/matrix notation and finish this up.

$$x = \begin{bmatrix} X_B \\ X_C \end{bmatrix}, c = \begin{bmatrix} -25 \\ -30 \end{bmatrix}$$

And now the constraints:

$$A = \begin{bmatrix} \frac{1}{200} & \frac{1}{140} \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, b = \begin{bmatrix} 40 \\ 6000 \\ 4000 \\ 0 \\ 0 \end{bmatrix}$$

And so now in canonical form we have:

$$\begin{aligned}
\min \ & c^T x \\
\text{s.t.} \ & Ax \leq b
\end{aligned}$$

Standard Form

This modification isn't too bad going from canonical form now. We need slack variables to handle the inequalities but nothing too crazy.

Essentially, all we gotta do is create a slack variable s_i for all of the inequalities. These will be set up such that $s_i \geq 0$. These end up going with the non-negativity constraints on X_B, X_C, so we only need 3 slack variables in total. One for each of the main constraints.

For a simple example, the second inequality becomes $X_B + 0X_C + s_2 = 6000$.

$$\begin{aligned}
\frac{1}{200}X_B + \frac{1}{140}X_C + s_1 &= 40 \\
X_B + 0X_C + s_2 &= 6000 \\
0X_B + X_C + s_3 &= 4000 \\
X_B, X_C, s_1, s_2, s_3 &\geq 0
\end{aligned}$$

As these add new variables we adjust the matrices as such.

$$x = \begin{bmatrix} X_B & X_C & s_1 & s_2 & s_3 \end{bmatrix}^T$$

$$c = \begin{bmatrix} -25 & -30 & 0 & 0 & 0 \end{bmatrix}^T$$

$$A = \begin{bmatrix} \frac{1}{200} & \frac{1}{140} & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 40 \\ 6000 \\ 4000 \end{bmatrix}$$

So now we have everything. In standard form we have:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

Homework 2

Given the system of equations, remove a set of 2 variables aside from $\{x_1, x_4\}$.

$$x_1 + 2x_2 + 3x_3 = 6$$
$$x_1 + x_2 + x_3 + x_4 = 4$$
$$x_1, x_2, x_3, x_4 \geq 0$$

For this homework we will remove $\{x_2, x_4\}$.

Step 1: Solve for x_2.

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 6 \\ 2x_2 &= 6 - 1x_1 - 3x_3 \\ x_2 &= 3 - \frac{1}{2}x_1 - \frac{3}{2}x_3 \end{aligned}$$

Step 2: Solve for x_4. Plug in x_2.

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 &= 4 \\ x_1 + 3 - \frac{1}{2}x_1 - \frac{3}{2}x_3 + x_3 + x_4 &= 4 \\ 3 + \frac{1}{2}x_1 - \frac{1}{2}x_3 + x_4 &= 4 \\ x_4 &= 1 + \frac{1}{2}x_3 - \frac{1}{2}x_1 \end{aligned}$$

Step 3: Handle non-negativity constraint. Simplify.

$$\begin{aligned} 3 - \frac{1}{2}x_1 - \frac{3}{2}x_3 &\geq 0 \\ -\frac{1}{2}x_1 - \frac{3}{2}x_3 &\geq -3 \\ \frac{1}{2}x_1 + \frac{3}{2}x_3 &\leq 3 \\ x_1 + 3x_3 &\leq 6 \end{aligned}$$

$$\begin{aligned} 1 + \frac{1}{2}x_3 - \frac{1}{2}x_1 &\geq 0 \\ x_1 - x_3 &\leq 2 \end{aligned}$$

So our new setup is:

$$x_1 + 3x_3 \leq 6$$
$$x_1 - x_3 \leq 2$$
$$x_1, x_3 \geq 0$$

Homework 3

Problem Decide for our running example for which combinations of basic variables we get a basic feasible or infeasible solution via a computation. (From lecture 4).

I'll be using the same system of equations as in homework 2.

$$A = \begin{vmatrix} 1 & 2 & 3 & 0 \\ 1 & 1 & 1 & 1 \end{vmatrix}, \quad b = \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

To get our basic solution we need to take A and take 2 subsets of columns from it, B and N. The columns we choose for these don't need to be consecutive or anything. For the sake of this assignment, we'll nab 2 columns from A to use for B and the other two will go to N.

We'll be ignoring N after that as it's set to 0.

Our basic solution will be calculated as such:

$$x_B = B^{-1}b$$

To choose something different from the lecture, I'll be using columns 2 and 3 for B. N then will get columns 1 and 4.

$$B = \begin{vmatrix} 2 & 3 \\ 1 & 1 \end{vmatrix}, \quad N = \begin{vmatrix} 1 & 0 \\ 1 & 1 \end{vmatrix}$$

From there it is as simple as doing the calculation.

$$\begin{aligned} x_B &= B^{-1}b && (1) \\ &= \begin{vmatrix} -1 & 3 \\ 1 & -2 \end{vmatrix} \begin{pmatrix} 6 \\ 4 \end{pmatrix} && (2) \\ &= \begin{pmatrix} 6 \\ -2 \end{pmatrix} && (3) \\ & && (4) \end{aligned}$$

So this is giving us the basic solution of $x_2 = 6, x_3 = -2$. Because $-2 < 0$, this contradicts the non-negativity constraint for x_3. Thus, this represents an infeasible solution for the system.

Homework 4: AMPL Book Exercise 1-2

Problem: The steel model for this chapter can be further modified to reflect various changes in production requirements. For each part below, explain the modifications to Figures 1-6a and 1-6b that would be required to achieve the desired changes. Make each change in isolation, not carrying modifications from part to part.

Reference Information

Before we begin, let's just keep the default info and solution up here as an easy reference.

Files Figures 1-6a and 1-6b both use the **steel4** .dat and .mod files. So I'll be using them as a base and modifying them.

The given steel linear program has the following solution:

$$\text{Total Profit} \approx 190071.43$$
$$\text{Bands} \approx 3357.14$$
$$\text{Coils} = 500$$
$$\text{Plates} \approx 3142.86$$

As for time used, we use 35 hours on the reheat stage and 40 hours on the roll stage.

A

Problem: How would you change the constraints so that total hours used by all products must equal the total hours available for each stage? Solve the linear program and verify that you get the same results. Why is there no difference in solution?

Solution: All we need to do here is modify one line. Line 18 specifically. We change the $\leq$ to a strict $=$.

```
subject to Time: sum {p in PROD} (1/rate[p,s]) * Make[p] =
    avail[s];
```

The solution it gives is the exact same. This is because our goal is to produce as much as we can to maximize profit. So the original solution is already using up all of the available hours. We can check this programmatically and check the hours both solutions used. I don't have that included in here, but I personally verified that this was the case.

B

Problem: How would you add to the model to restrict the total weight of all products to be less than a new parameter, max_weight? Solve the linear program for a weight limit of 6500 tons, and explain how this extract restriction changes the results.

Solution: We need to make a few modifications here. We'll need to edit both the .dat and .mod files. In .dat we simply add a new parameter alongside our availability parameter.

```
param avail := reheat 35 roll 40;
param max_weight := 6500;
```

In the .mod file we adjust two things. First we read in that max weight parameter and then include in as a new constraint at the bottom.

```
param max_weight >= 0   # Total tons of weight allowed
    across all products

subject to Total_Weight:
  sum{p in PROD} Make[p] <= max_weight;
```

$$\text{Total Profit} \approx 183791.67$$
$$\text{Bands} \approx 1541.67$$
$$\text{Coils} = 1458.33$$
$$\text{Plates} \approx 3500$$

We also change up our usage of time a bit. Spending 32.5 hours on reheat and 40 on rolling.

This restriction forces us to think about profit per ton beyond just rate. We now have to consider weight limits and how that effects profits. The amount of coils we make here skyrockets because because they are the lightest of the products we can produce.

C

Problem: How would you change the objective function to maximize total tons? Does this make a difference to the solution?

Solution: This is the simplest to change. Just remove the profit from the objective function as it already factors in weight.

```
  maximize Total_Weight: sum {p in PROD} Make[p];
```

Funnily enough this ends up with the exact same results as the original model!

$$\begin{aligned} \text{Total Weight} &= 7000 \\ \text{Bands} &\approx 3357.14 \\ \text{Coils} &= 500 \\ \text{Plates} &\approx 3142.86 \end{aligned}$$

We just don't get our profit shown in the objective function is all as it shows the total weight. The profit is the exact same as well, it just isn't shown here.

D

Problem: Suppose that instead of the lower bounds represented by `commit[p]` in our model, we want to require that each product represent a certain share of the total tons produced. In the algebraic notation of Figure 1-1, this new constraint might be represented as

$$X_j \geq s_j \sum_{k \in P} X_k, \text{ for each } j \in P$$

where s_j is the minimum share associated with project j. How would you change the AMPL model to use this constraint in place of the lower bounds `commit[p]`? If the minimum shares are 0.4 for bands and plate, and 0.1 for coils, what is the solution?

Verify that if you change the minimum shares to 0.5 for bands and plate, and 0.1 for coils, the linear program gives an optimal solution that produces nothing, at zero profit. Explain why this makes sense.

Solution: This one is a bit more involved but honestly not too bad. We'll first change our data file. We want to remove the `commit` parameter and replace it with our new `share` parameter.

```
param:    profit  market  share :=
  bands     25     6000    0.4
  coils     30     4000    0.1
  plate     29     3500    0.4 ;
```

Moving onto our model file, we need to adjust a couple things. First, we replace the lower bound on `Make` that was previously set by `commit` with 0. Then we add a new constraint to the bottom of the file to ensure the shares are working correctly.

```
param share {PROD} >= 0;      # minimum proportion of total
    tons for each product

var Make {p in PROD} >= 0, <= market[p]; # tons produced

subject to Share {p in PROD}:
    Make[p] >= share[p] * sum {k in PROD} Make[k];
```

With this our new solution is as follows:

$$\text{Total Profit} = 189700$$
$$\text{Bands} = 3500$$
$$\text{Coils} = 700$$
$$\text{Plates} = 2800$$

Some quick sanity checks let us know all of these meet the minimum proportions we set. Also, if we change the shares in the data file to be 0.5, 0.1, 0.5 we do end up producing nothing as expected. This is because those shares are proportions and they total to 1.1 which is impossible. Those requirements can't be met so we produce nothing.

E

Problem: Suppose there is an additional finishing stage for plates only, with a capacity for 20 hours and a rate of 150 ton per hour. Explain how you could modify this data, without changing the model, to incorporate this new stage.
Solution: This is actually, thankfully, very easy to do! We can simply add a new stage to the data file and set arbitrarily large limits for bands and coils.

```
set STAGE := reheat roll finishing;

param rate:  reheat  roll  finishing:=
  bands        200    200   1e9
  coils        200    140   1e9
  plate        200    160   150;

param avail := reheat 35 roll 40 finishing 20;
```

This automatically gets worked in with 0 modifications to the model file!