Optimisation Summative Coursework

Jake Mortimer - wtxd25

February 6, 2020

1 Question 1

The Simplex Tableuas are as follows:

$$T^1 = \left[\begin{array}{c|ccccc|c} 1 & 1 & -3 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & -1 & 1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 2 & -1 & 0 & 0 & 1 & 3 \end{array}\right]$$

We choose the negative value in the first row with the smallest index; -3. In this column, we find $t = min(-, \frac{1}{1}, -) = 1$. $\therefore$ we pivot on the position $(3, 3)$.

$$T^2 = \left[\begin{array}{c|ccccc|c} 1 & -2 & 0 & 0 & 3 & 0 & 3 \\ \hline 0 & 0 & 0 & 1 & 1 & 0 & 2 \\ 0 & -1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 4 \end{array}\right]$$

We choose the negative value in the first row with the smallest index; -2. In this column, we find $t = min(-, -, \frac{4}{1}) = 4$. $\therefore$ we pivot on the position $(2, 4)$.

$$T^3 = \left[\begin{array}{c|ccccc|c} 1 & 0 & 0 & 0 & 5 & 2 & 11 \\ \hline 0 & 0 & 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 & 2 & 1 & 5 \\ 0 & 1 & 0 & 0 & 1 & 1 & 4 \end{array}\right]$$

We stop here because there are no more negative values in the first row, this means that the optimal solution has been found. The **optimal solution** is $\overline{x} = (4, 5, 2, 0, 0, 0)^T$, with value $\overline{z}(\overline{x}) = 11$. For the **original problem** the objective function would give $z(\overline{x}) = -11$.

2 Question 2

The Simplex Tableuas are as follows:

$$T^1 = \left[\begin{array}{c|cccccc|c} 1 & -2 & -1 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & -2 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 & 1 & 0 & 2 \\ 0 & 2 & -3 & -1 & 0 & 0 & 1 & 6 \end{array}\right]$$

We choose the negative value in the first row with the smallest index; -2. In this column, we find $t = min(-, \frac{2}{1}, \frac{6}{2}) = 2$. $\therefore$ we pivot on the position $(2, 3)$.

$$T^2 = \begin{bmatrix} 1 & 0 & -3 & 1 & 0 & 2 & 0 & 4 \\ \hline 0 & 0 & -1 & 1 & 1 & 2 & 0 & 5 \\ 0 & 1 & -1 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & -1 & -1 & 0 & -2 & 1 & 2 \end{bmatrix}$$

In this tableau, we only have one negative value in the first row; -3. The corresponding column is $(-1,-1,-1)^T$. With the values in the final column, the constraints, these put no bounds on the value of t. As such, the LP is **unbounded**.

3 Question 3

We capture the problem in a Linear Program (LP) as follows:
Maximise $(10, 15, 25, 25)\mathbf{x}$
Subject to:
$$\begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{3} \end{pmatrix} \mathbf{x} \le \begin{pmatrix} 10 \\ 5 \\ 11 \end{pmatrix}$$
with $\mathbf{x} \ge \mathbf{0}$

This is the correct formulation of the problem defined in Question 3. Each of the three rows represents the use of the Cyan, Magenta, and Yellow colours in the mixing process. The variables in the vector $\mathbf{x}$ are x_1, x_2, x_3, x_4 and these represent the number of red, green, blue and black, respectively, that we have produced. The objective function follows directly from the definition of the price per unit from each of the colours of paint. In the first row, we have 0 in the first column as cyan is not used in the making of red, $\frac{1}{2}$ in the second column as we produce 2 units of green for 1 unit of cyan. The same logic is for the third and fourth column values in the first row. This definition then extends down the rows. The limit on each of these is the maximum value of the cyan, magenta, and yellow colours of paint.
See the solution to this LP in q3.py.

4 Question 4

4.1 (a)

We capture the problem as a Integer Linear Program (IP) as follows:
Maximise $(60, 70, 40, 70, 16, 100)\mathbf{x}$
Subject to:
$$\begin{pmatrix} 6 & 7 & 4 & 9 & 3 & 8 \end{pmatrix} \mathbf{x} \le \begin{pmatrix} 20 \end{pmatrix}$$
with $\mathbf{0} \le \mathbf{x} \le \mathbf{1}$. $\mathbf{x}$ is the vector (A, B, C, D, E, F).

In the problem description we are led to believe that there is only one of each item. This allows us to use an IP over the values 0 and 1 to formulate this

problem. We are using one variable for each item, and all the variables are contained in the column vector $\mathbf{x}$. The objective function can be taken exactly from the description of the problem; the coefficient of each of the variables in $\mathbf{x}$ is the value of the item, and we are seeking to maximise this objective function. There is only one constraint on this problem which is that the linear combination of weights, where the coefficients of these weights is the values of the variables corresponding to the items, must be less than or equal to 20.

See the solution to this LP in q4a.py.

4.2 (b)

We capture the problem as a Integer Linear Program (IP) as follows:

Maximise $(60, 70, 40, 70, 16, 100)\mathbf{x}$

Subject to:

$$\begin{pmatrix} 6 & 7 & 4 & 9 & 3 & 8 \\ 0 & 0 & 1 & -1 & 0 & 0 \end{pmatrix} \mathbf{x} \le \begin{pmatrix} 20 \\ 0 \end{pmatrix}$$

with $\mathbf{0} \le \mathbf{x} \le \mathbf{1}$. $\mathbf{x}$ is the vector (A, B, C, D, E, F).

This problem is the same formulation as above except now there is a dependency between C and D; if we have C then must also have D, but if we have D then we don't need to have C. As such the value of $D \ge C$. This can be rewritten as $C - D \le 0$, this can be added to the list of constraints in the IP to give the required results.

See the solution to this LP in q4b.py.

4.3 (c)

We capture the problem as a Integer Linear Program (IP) as follows:

Maximise $(60, 70, 40, 70, 16, 100, -15)\mathbf{x}$

Subject to:

$$\begin{pmatrix} 6 & 7 & 4 & 9 & 3 & 8 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \mathbf{x} \le \begin{pmatrix} 20 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

with $\mathbf{x} \ge \mathbf{0}$. $\mathbf{x}$ is the vector (A, B, C, D, E, F, G).

In this problem we can have additional weight on top of the 20kg for £15 per kg over the limit. The number of kg we go over the limit is modelled by the new variable G. In the objective function it's coefficient is -15; modelling the £15 fine for each kg over the limit. We can model the weight allowed constraint as follows: $6A + 7B + 4C + 9D + 3E + 8F \le 20 + G$. We then subtract G from both sides to get the first row of the constraints matrix. The remaining rows specify that all variables, except G, in $\mathbf{x}$ are less than or equal to 1. We cannot simply do this as before with "$\mathbf{0} \le \mathbf{x} \le \mathbf{1}$" as G can be greater than 1.

See the solution to this LP in q4c.py.