The CopyMaster problem is solved using the MIP

\[
\begin{align*}
\min_{x, y, T} & \quad Z = T \\
\text{s.t.} & \quad \sum_{i=1}^{6} c_{ij} x_{ij} - T \leq 0 \quad i = 1, 2, 3, 4 \\
& \quad \sum_{i=1}^{6} x_{ij} = n_j \quad j = 1, 2, 3, 4, 5, 6 \\
& \quad -x_{ij} + 24 y_{ij} \leq 0 \quad \left\{ \begin{array}{l} i = 1, 2, 3, 4 \\ j = 1, 2, 3, 4, 5, 6 \end{array} \right. \\
& \quad x_{ij}, y_{ij} \in \mathbb{Z}, x \geq 0 \\
& \quad T \in \mathbb{R}
\end{align*}
\]

This MIP has 49 decision variables and 6 equality constraints, 52 inequality constraints, and various box constraints.

These notes describe how to represent this problem in Matlab standard form

\[
\begin{align*}
\min_{w} & \quad Z = c^T w \\
\text{s.t.} & \quad A x \leq b \\
& \quad a_0 x = b_e \\
& \quad l \leq x \leq u \end{align*}
\]

\[
\begin{align*}
\min_{w} & \quad Z = c^T w \\
\text{s.t.} & \quad A w \leq b \\
& \quad A_e w = b_e \\
& \quad l \leq w \leq u \\
& \quad w \in \mathbb{R} \text{ (or } \mathbb{Z} \text{)}
\end{align*}
\]
The first key idea is that the set of decision variables must be represented as a vector in $\mathbb{R}^n$ (in this case, $\mathbb{R}^{49}$). Because the variables $x_{ij}$ and $y_{ij}$ are not naturally represented as vectors we must declare a variable order. I will choose

$$\omega = [x_{11}, x_{21}, x_{31}, x_{41}, x_{12}, x_{22}, \ldots, x_{46}, y_{11}, y_{21}, y_{31}, y_{41}, y_{12}, y_{22}, \ldots, y_{46}, T]^T$$

This variable order establishes the construction details of the various matrices/vectors in the problem ($A, b, c, \text{etc.}$). For example, we now have

$$c = \left[ \underbrace{0, 0, 0, \ldots, 0, 0, 0}_{48 \text{ zeros}}, 1 \right]^T$$

so that $c^T \omega = T$.

We can construct the equality constraint matrices as follows. The first constraint (for $j = 1$) is

$$x_{11} + x_{21} + x_{31} + x_{41} = N_1$$

which corresponds to a row in $A_e$ that is:

$$\left[ \overbrace{1, 1, 1, 1}^{4 \text{ ones}}, \underbrace{0, 0, \ldots, 0, 0}_{45 \text{ zeros}} \right]$$

and an element of $b_e$ which is $N_1 = 641$.
All six constraints together look like this:

\[ A_e = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 25 \text{ more columns of all zeros} \end{bmatrix} \]
\[ b_e = \begin{bmatrix} 641 \\ 190 \\ 386 \\ 469 \\ 395 \\ 482 \end{bmatrix} \]

\( A_e \) is 6x49 because there are six equality constraints and 49 decision variables.

There are box constraints: \( x \geq 0, \ 0 \leq y \leq 1 \) which we can write together as

\[ w = \begin{bmatrix} x \\ y \\ T \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \leq \begin{bmatrix} x \\ y \\ T \end{bmatrix} \leq \begin{bmatrix} M \\ 1 \\ \text{inf} \end{bmatrix} \]

So, if \( l \leq w \leq u \), we have

\[ l = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix}^T \quad \text{49 zeros} \]

\[ u = \begin{bmatrix} m & m & \cdots & m & 1 & 1 & \cdots & 1 & \text{inf} \end{bmatrix}^T \quad \text{24 m's} \quad \text{24 1's} \quad \text{one infinity} \]

All other matrices and vectors can be constructed similarly.
There are many useful Matlab/Octave commands that can be used to construct large matrices that have a lot of structure.

- `ones(a,b)` creates an array of `a` rows and `b` columns filled with ones.
- `zeros(a,b)` same idea but fill with zeros.
- `d*ones(a,b)` same idea but fill with value `d`.

So, in our example problem

```matlab
C = [zeros(48,1); 1];
```

48 rows of zeros, add a new last row is a 1.

```matlab
l = zeros(49,1);
```

```matlab
W = W*ones
```

```matlab
u = [641*ones(24,1); ones(24,1); inf];
```
\texttt{eye(n)} creates the \( n \times n \) identity matrix
\[
\text{ex: } \texttt{eye}(3) = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}.
\]

\texttt{repmat(v, a, b)} creates an array of numbers by repeating array \( v \) \( a \) times vertically and \( b \) times horizontally.

\text{ex: } \texttt{repmat([1 2], 2, 3)}
\[
= \begin{bmatrix}
1 & 2 & 1 & 2 & 1 & 2 \\
1 & 2 & 1 & 2 & 1 & 2
\end{bmatrix}.
\]

\texttt{kron(u, v)} creates an array by multiplying \( v \) by every element of \( u \). \( v \) can also be an array.

\text{ex: } \texttt{kron(eye(2), 4)} = \begin{bmatrix}
4 & 0 \\
0 & 4
\end{bmatrix}
\]

\text{ex: } \texttt{kron(eye(2), [3 2])} = \begin{bmatrix}
3 & 2 & 0 & 0 \\
0 & 0 & 3 & 2
\end{bmatrix}
\]

\text{ex: } \texttt{kron([1 2], [3 4])} = \begin{bmatrix}
3 & 4 & 6 & 8
\end{bmatrix}.
Now we can construct matrix $A_e$.

\[
A_1 = \text{kron}(\text{eye}(6), \text{ones}(1,4)) ; \\
A_2 = \text{zeros}(6,25) ; \\
A_e = [A_1 \ A_2] ;
\]