Amadeus Job Interview Task

Network Table Generation Method, which supplements the optimization algorithm outlined in the main documentation

By: Oscar A. Nieves

March 2022

Suppose we have an input array in which each entry is a list of machines which are available for sale on that particular day and looks like this:

Input array:

$$\left[\begin{Bmatrix} M_1 \\ M_2 \end{Bmatrix}, \begin{Bmatrix} M_3 \\ M_4 \end{Bmatrix}, \{\}, \{M_5\}, \{\} \right]$$

The idea is to create a network connection table that goes from left to right, as shown in the next slide:

$$\left\{\begin{matrix}
[M_1, M_1', M_1', M_1', M_1'] \\
[M_1, M_1', M_1', M_5, M_5'] \\
[M_1, M_3, M_3', M_3', M_3'] \\
[M_1, M_3, M_3', M_5, M_5'] \\
[M_1, M_4, M_4', M_4', M_4'] \\
[M_1, M_4, M_4', M_5, M_5'] \\
[M_2, M_2', M_2', M_2', M_2'] \\
[M_2, M_2', M_2', M_5, M_5'] \\
[M_2, M_3, M_3', M_3', M_3'] \\
[M_2, M_3, M_3', M_5, M_5'] \\
[M_2, M_4, M_4', M_4', M_4'] \\
[M_2, M_4, M_4', M_5, M_5']
\end{matrix}\right\}$$

The network table contains a set number of rows and columns. The number of columns matches D+1 (number of restructuring period days + 1). The number of rows corresponds to all plausible network paths we can create.

The elements marked with an apostrophe (‘) are modified in some way, by some function f(M) = M’.

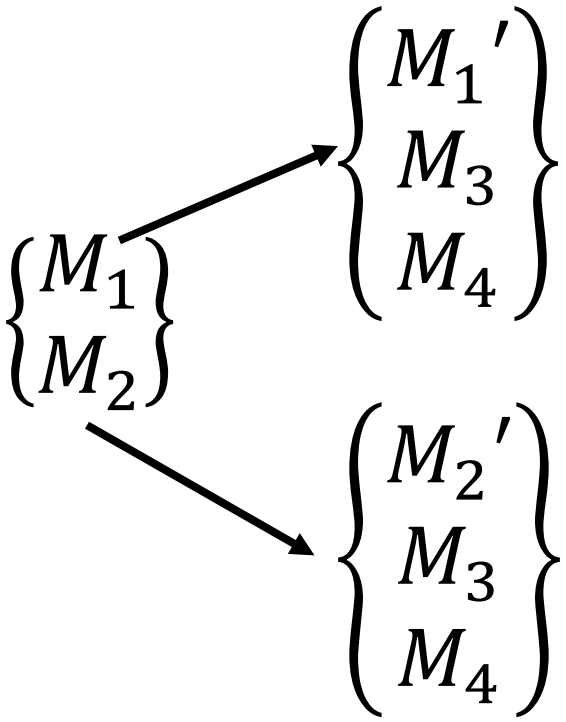
Once the network table is generated, the computation of the weights w and total amount of money at D+1 for all paths is very straight-forward, as it only needs to be done for each row, and assigns those final values to a column vector.

The optimizer then just grabs whatever the maximum value is in that final column vector.

Step 1

$$\left[\begin{Bmatrix} M_1 \\ M_2 \end{Bmatrix}, \begin{Bmatrix} M_3 \\ M_4 \end{Bmatrix}, \{\}, \{M_5\}, \{\}\right]$$

For each list M_i in the first array, duplicate the next array and add a modified copy of each element in current array like this:

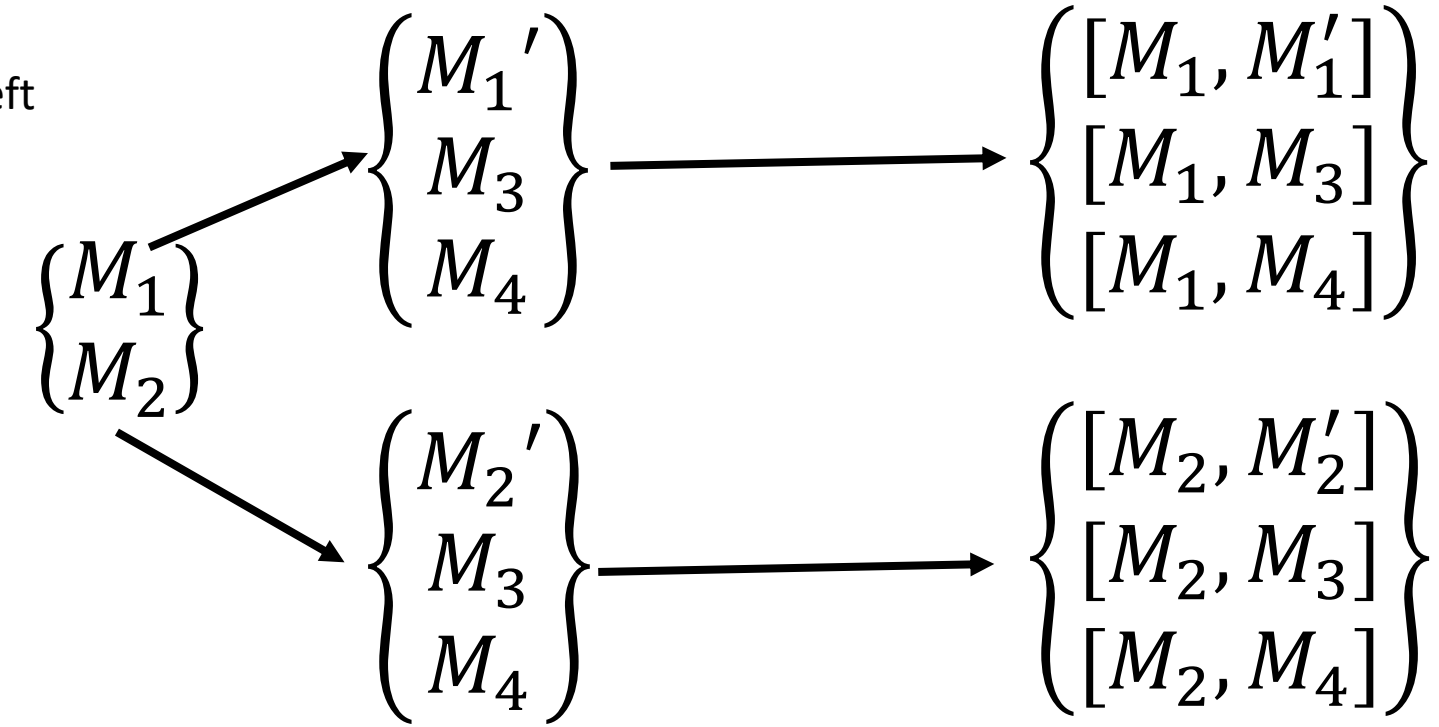


The dash ‘ means we are applying some function to M_1, for instance: $f(M_1) = M_1'$, it could be a simple operation such as changing one of the values in the list

Step 2

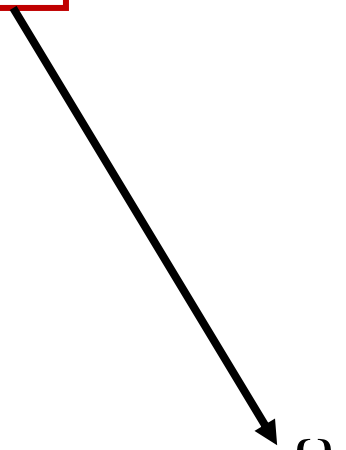
$$\left[\begin{Bmatrix} M_1 \\ M_2 \end{Bmatrix}, \begin{Bmatrix} M_3 \\ M_4 \end{Bmatrix}, \{\}, \{M_5\}, \{\}\right]$$

Write a sequence from left to right starting like this:



Step 3

$$\left[\begin{Bmatrix} M_1 \\ M_2 \end{Bmatrix}, \begin{Bmatrix} M_3 \\ M_4 \end{Bmatrix}, \{\}, \{M_5\}, \{\} \right]$$



$$\{\}$$

Using this new array, move unto the next pair:

$$\begin{Bmatrix} [M_1, M_1'] \\ [M_1, M_3] \\ [M_1, M_4] \\ [M_2, M_2'] \\ [M_2, M_3] \\ [M_2, M_4] \end{Bmatrix}$$

Step 4

$$\left[\begin{Bmatrix} M_1 \\ M_2 \end{Bmatrix}, \begin{Bmatrix} M_3 \\ M_4 \end{Bmatrix}, \boxed{\{\}}, \{M_5\}, \{\} \right]$$

If the next array is empty, we simply append a modified version (function operated on) of the last element in the previous arrays. For example here:

$$\begin{Bmatrix} [M_1, M_1'] \\ [M_1, M_3] \\ [M_1, M_4] \\ [M_2, M_2'] \\ [M_2, M_3] \\ [M_2, M_4] \end{Bmatrix} \rightarrow \begin{Bmatrix} [M_1, M_1', M_1'] \\ [M_1, M_3, M_3'] \\ [M_1, M_4, M_4'] \\ [M_2, M_2', M_2'] \\ [M_2, M_3, M_3'] \\ [M_2, M_4, M_4'] \end{Bmatrix}$$

Step 5 $\left[\begin{Bmatrix} M_1 \\ M_2 \end{Bmatrix}, \begin{Bmatrix} M_3 \\ M_4 \end{Bmatrix}, \{\}, \boxed{\{M_5\}}, \{\} \right]$

Repeat the same process over and over again. Remember: when the next input array is NOT empty, we need to create one copy of it for each array in our big matrix M like this

$$\begin{Bmatrix} [M_1, M_1', M_1'] \\ [M_1, M_3, M_3'] \\ [M_1, M_4, M_4'] \\ [M_2, M_2', M_2'] \\ [M_2, M_3, M_3'] \\ [M_2, M_4, M_4'] \end{Bmatrix} \longrightarrow \begin{Bmatrix} [M_1, M_1', M_1', M_1'] \\ [M_1, M_1', M_1', M_5] \\ [M_1, M_3, M_3', M_3'] \\ [M_1, M_3, M_3', M_5] \\ [M_1, M_4, M_4', M_4'] \\ [M_1, M_4, M_4', M_5] \\ [M_2, M_2', M_2', M_2'] \\ [M_2, M_2', M_2', M_5] \\ [M_2, M_3, M_3', M_3'] \\ [M_2, M_3, M_3', M_5] \\ [M_2, M_4, M_4', M_4'] \\ [M_2, M_4, M_4', M_5] \end{Bmatrix}$$

Here, the red element was duplicated from the last element in each array […] in the previous matrix M (and also modified by the function f(x) = x'), and the green element is what we added from the new array in our Input

Step 6

$$\left[\begin{Bmatrix} M_1 \\ M_2 \end{Bmatrix}, \begin{Bmatrix} M_3 \\ M_4 \end{Bmatrix}, \{\}, \{M_5\}, \{\} \right]$$

Again, move on to the next array in the input. If it is empty, simply duplicate each last element in the previous matrix:

$$\begin{Bmatrix} [M_1, M_1', M_1', M_1'] \\ [M_1, M_1', M_1', M_5] \\ [M_1, M_3, M_3', M_3'] \\ [M_1, M_3, M_3', M_5] \\ [M_1, M_4, M_4', M_4'] \\ [M_1, M_4, M_4', M_5] \\ [M_2, M_2', M_2', M_2'] \\ [M_2, M_2', M_2', M_5] \\ [M_2, M_3, M_3', M_3'] \\ [M_2, M_3, M_3', M_5] \\ [M_2, M_4, M_4', M_4'] \\ [M_2, M_4, M_4', M_5] \end{Bmatrix} \longrightarrow \begin{Bmatrix} [M_1, M_1', M_1', M_1', M_1'] \\ [M_1, M_1', M_1', M_5, M_5'] \\ [M_1, M_3, M_3', M_3', M_3'] \\ [M_1, M_3, M_3', M_5, M_5'] \\ [M_1, M_4, M_4', M_4', M_4'] \\ [M_1, M_4, M_4', M_5, M_5'] \\ [M_2, M_2', M_2', M_2', M_2'] \\ [M_2, M_2', M_2', M_5, M_5'] \\ [M_2, M_3, M_3', M_3', M_3'] \\ [M_2, M_3, M_3', M_5, M_5'] \\ [M_2, M_4, M_4', M_4', M_4'] \\ [M_2, M_4, M_4', M_5, M_5'] \end{Bmatrix}$$

$$\left[\begin{Bmatrix} M_1 \\ M_2 \end{Bmatrix}, \begin{Bmatrix} M_3 \\ M_4 \end{Bmatrix}, \{\}, \{M_5\}, \{\}\right]$$

How to know if the size of the final output array is correct? We look at the input array first, then we count how many elements it has (horizontally). In this case, there are 5 columns, so our output array will also have 5 columns.

The next step is to look at the first element in the input. It has 2 elements, so it has size 2. We will multiply the size of this by the size of the next element + 1. Since the second element has 2 elements, then we will say it has size 2+1 = 3. We repeat the same process for every other element in the input. Arrays that are empty {} have size 0, however since we are adding 1, they have new size 1. In this case, the total product is:

2(2+1)(0+1)(1+1)(0+1) = 2(3)(1)(2)(1) = 12

So the final output array must have 12 Rows

$$\begin{Bmatrix}
[M_1, M_1', M_1', M_1', M_1'] \\
[M_1, M_1', M_1', M_5, M_5'] \\
[M_1, M_3, M_3', M_3', M_3'] \\
[M_1, M_3, M_3', M_5, M_5'] \\
[M_1, M_4, M_4', M_4', M_4'] \\
[M_1, M_4, M_4', M_5, M_5'] \\
[M_2, M_2', M_2', M_2', M_2'] \\
[M_2, M_2', M_2', M_5, M_5'] \\
[M_2, M_3, M_3', M_3', M_3'] \\
[M_2, M_3, M_3', M_5, M_5'] \\
[M_2, M_4, M_4', M_4', M_4'] \\
[M_2, M_4, M_4', M_5, M_5']
\end{Bmatrix}$$