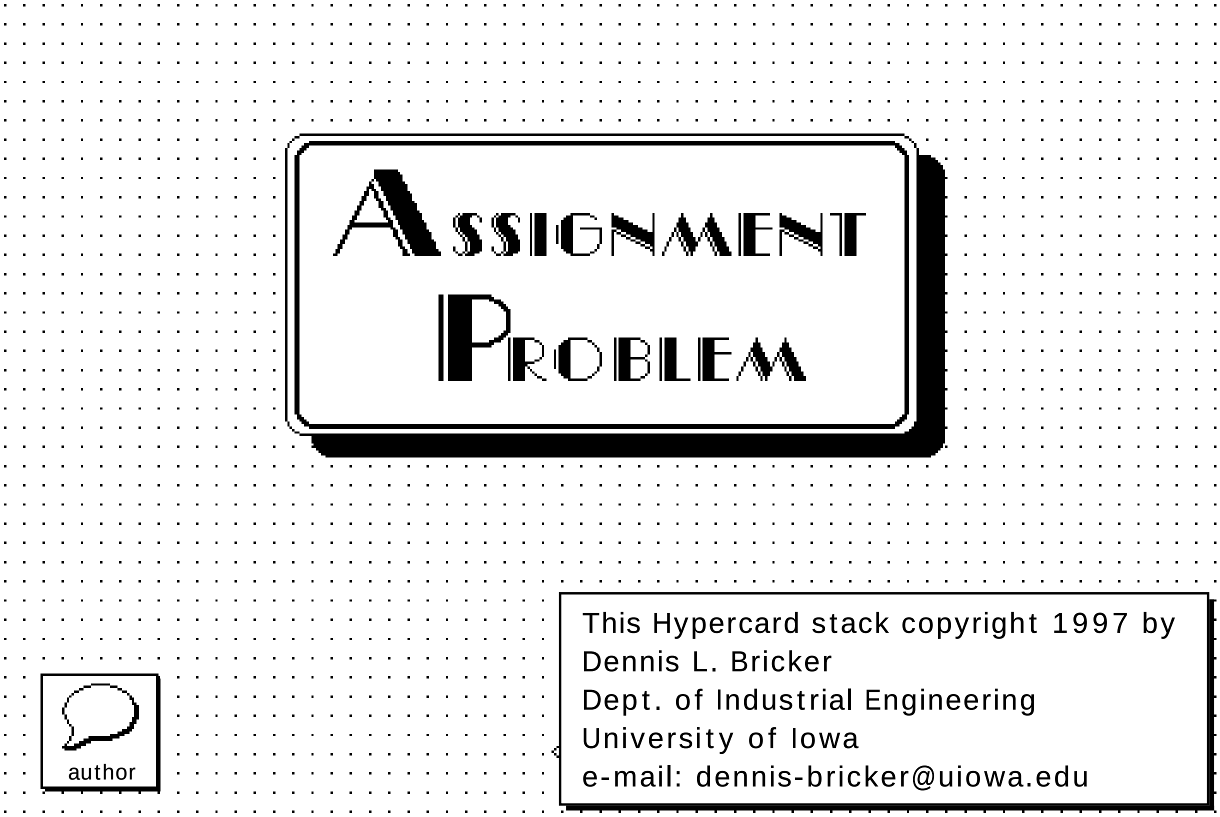


Assignment Problem

This Hypercard stack copyright 1997 by
Dennis L. Bricker
Dept. of Industrial Engineering
University of Iowa
e-mail: dennis-bricker@uiowa.edu

☞ Linear Assignment Problem

☞ Quadratic Assignment Problem

☞ Generalized Assignment Problem

Assignment Problem

		jobs				
		A	B	C	D	E
machines	1	5	3	2	3	4
	2	6	2	1	4	3
	3	4	3	3	2	2
	4	5	4	2	5	2
	5	3	3	2	4	3

cost of completing job

What is the least-cost way of assigning a machine to each of 5 jobs (one job/machine)?

D.L. Bricker

THE ASSIGNMENT PROBLEM

Each of **n** ***resources*** must be assigned to one of **n** ***activities***, and each activity is assigned exactly one resource.

A cost C_{ij} results if resource ***i*** is assigned to activity ***j***.

The objective is to minimize the total cost of assigning every resource to an activity.

Example: assigning jobs to machines in a job-shop

IP formulation

Let $$X_{ij} = \begin{cases} 1 & \text{if resource i is assigned to activity j} \\ 0 & \text{otherwise} \end{cases}$$

AP

Minimize $\sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij}$

subject to

$$\sum_{j=1}^{n} X_{ij} = 1 \text{ for } i=1, 2, \ldots n$$

each resource is assigned to exactly one activity

$$\sum_{i=1}^{n} X_{ij} = 1 \text{ for } j=1, 2, \ldots n$$

each activity is assigned exactly one resource

$$X_{ij} \,\varepsilon\, \{0,1\} \text{ for all } i \,\&\, j$$

AP

Minimize $\sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij}$

subject to

$$\sum_{j=1}^{n} X_{ij} = 1 \text{ for } i=1, 2, \ldots n$$

$$\sum_{i=1}^{n} X_{ij} = 1 \text{ for } j=1, 2, \ldots n$$

$$X_{ij} \, \varepsilon \, \{0,1\} \text{ for all } i \text{ \& } j$$

Note that this is a special case of the transportation problem (with supplies & demands each equal to 1)!

If the restriction that X is 0 or 1 is replaced with a nonnegativity restriction, the LP solution will still be integer!

D.L. Bricker

AP

$$\text{Minimize } \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij}$$

subject to

$$\sum_{j=1}^{n} X_{ij} = 1 \text{ for } i=1, 2, \ldots n$$

$$\sum_{i=1}^{n} X_{ij} = 1 \text{ for } j=1, 2, \ldots n$$

$$X_{ij} \varepsilon \{0,1\} \text{ for all } i \;\&\; j$$

number of basic variables is 2n-1.

number of positive variables is n

Although AP could be solved by the simplex method for TP, all the basic solutions are highly degenerate, which lessens the efficiency of the algorithm.

Properties of the Assignment Problem

For each i, exactly one assignment $X_{ij}=1$ is made
For each j, exactly one assignment $X_{ij}=1$ is made

Therefore,

If a number **δ** is added to (or subtracted from) every cost in a certain row (or column) of the matrix C,
then every feasible set of assignments will have its cost increased (or decreased) by **δ**,
and the optimal set of assignments remains optimal!

For example, if we add **δ** *to row 1, the total cost is increased by*

$$\sum_{j=1}^{n} \delta X_{1j} = \delta \sum_{j=1}^{n} X_{1j} = \delta$$ *(independent of X!)*

Properties of the Assignment Problem

If all costs C_{ij} are nonnegative, and if there is a set of assignments with total cost equal to zero,

then

that set of assignments must be optimal.

The "Hungarian Method" solves the assignment problem by adding &/or subtracting quantities in rows &/or columns until an assignment with zero cost is found.

Example

Four machines are available to process four jobs.

The processing time for each machine/job assignment is as follows:

		job			
		1	2	3	4
machine	A	4	6	5	5
	B	7	4	5	6
	C	4	7	6	4
	D	5	3	4	7

What is the assignment (one job per machine) which will minimize total processing time?

Row reduction

For example, 4 is subtracted from each cost in the first row:

$$\begin{array}{cc|cccc} & & & \text{job} & & \\ & & 1 & 2 & 3 & 4 \\ \hline & A & 4 & 6 & 5 & 5 \\ \text{machine} & B & 7 & 4 & 5 & 6 \\ & C & 4 & 7 & 6 & 4 \\ & D & 5 & 3 & 4 & 7 \end{array} \Rightarrow \begin{array}{cc|cccc} & & & \text{job} & & \\ & & 1 & 2 & 3 & 4 \\ \hline & A & 0 & 2 & 1 & 1 \\ \text{machine} & B & 3 & 0 & 1 & 2 \\ & C & 0 & 3 & 2 & 0 \\ & D & 2 & 0 & 1 & 4 \end{array}$$

From each row, subtract the smallest cost.
This introduces at least one zero into each row!

Column reduction

Only column 3 lacks a zero, so only column 3 is reduced:

$$\text{machine}\ \begin{array}{c|cccc} & \multicolumn{4}{c}{\text{job}}\\ & 1 & 2 & 3 & 4\\ \hline A & 0 & 2 & 1 & 1\\ B & 3 & 0 & 1 & 2\\ C & 0 & 3 & 2 & 0\\ D & 2 & 0 & 1 & 4 \end{array} \Rightarrow \text{machine}\ \begin{array}{c|cccc} & \multicolumn{4}{c}{\text{job}}\\ & 1 & 2 & 3 & 4\\ \hline A & 0 & 2 & 0 & 1\\ B & 3 & 0 & 0 & 2\\ C & 0 & 3 & 1 & 0\\ D & 2 & 0 & 0 & 4 \end{array}$$

From each column, subtract the smallest cost.
If a column already has a zero, it is unchanged.
Otherwise, a zero is introduced into the column.

$$
\begin{array}{c|cccc}
 & & & \textit{job} & \\
\textit{machine} & 1 & 2 & 3 & 4 \\
\hline
A & \boxed{0} & 2 & 0 & 1 \\
B & 3 & \boxed{0} & 0 & 2 \\
C & 0 & 3 & 1 & \boxed{0} \\
D & 2 & 0 & \boxed{0} & 4
\end{array}
$$

Examining the cost matrix, we can find an assignment with total cost equal to zero:

machine	*job*
A	1
B	2
C	4
D	3

Therefore, this must be an optimal assignment!

Sometimes, however, one cannot find a zero-cost assignment after row- & column-reduction.

For example: machine C cannot be assigned to both jobs 1 & 4, so one job must be assigned a machine with positive cost

job

machine

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline A & 4 & 2 & 0 & 1 \\ B & 3 & 0 & 0 & 2 \\ C & \boxed{0} & 3 & 1 & \boxed{0} \\ D & 2 & 0 & 0 & 4 \end{array}$$

Hungarian Algorithm

Step 0 Convert to standard form, with # rows = # columns

Step 1 *Row reduction:* find the smallest cost in each row, and reduce all costs in that row by this amount.

Step 2 *Column reduction:* find the smallest cost in each column, and reduce all costs in the column by this amount.

D.L. Bricker

Hungarian Algorithm

Step 3 find the minimum number of lines through rows &/or columns necessary to cover all of the zeroes in the cost matrix. If this equals n, STOP.

Step 4 locate the smallest unlined cost, $\bar{c}$. Subtract this cost from all unlined costs, and add to costs at intersections of lines. Return to step 3.

D.L. Bricker

Justification for step 4:

"Subtract smallest unlined cost $\overline{c}$ from all unlined costs; add to costs at intersections of lines."

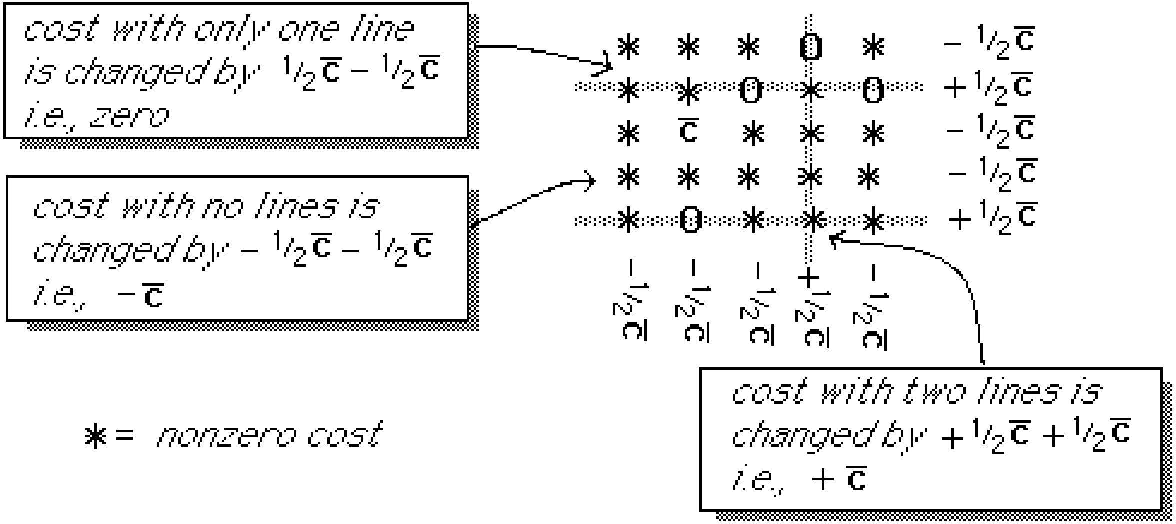
is equivalent to

"Subtract $\frac{1}{2}\overline{c}$ from each unlined row & each unlined column.

Add $\frac{1}{2}\overline{c}$ to each lined row and each lined column."

"Subtract $\frac{1}{2}\overline{c}$ from each unlined row & each unlined column.

Add $\frac{1}{2}\overline{c}$ to each lined row and each lined column."



cost with only one line is changed by $1/2\,\overline{\mathbf{c}} - 1/2\,\overline{\mathbf{c}}$ *i.e., zero*

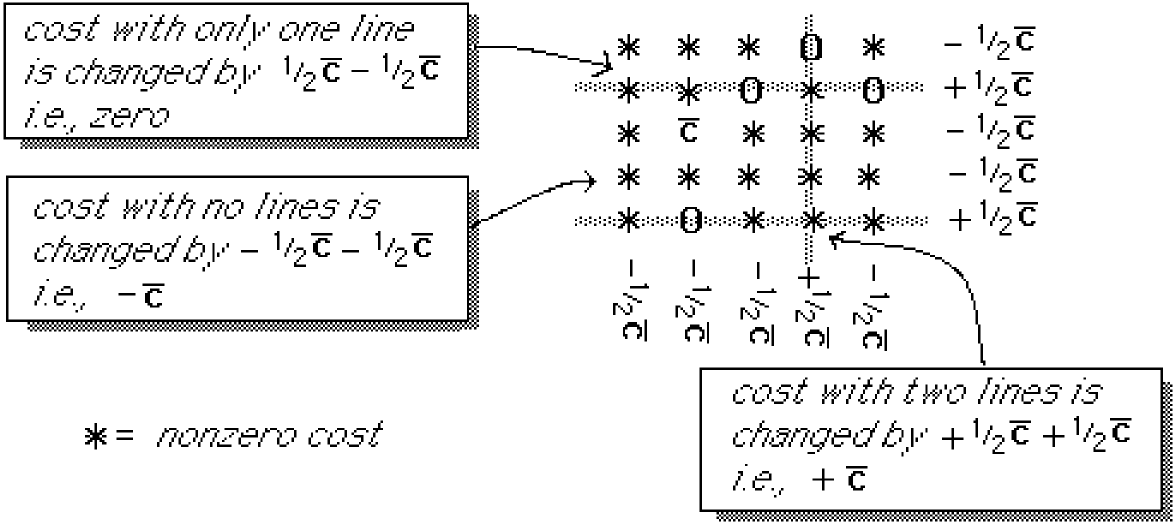
cost with no lines is changed by $-1/2\,\overline{\mathbf{c}} - 1/2\,\overline{\mathbf{c}}$ *i.e.,* $-\overline{\mathbf{c}}$

$$\begin{array}{ccccc|c}
\ast & \ast & \ast & 0 & \ast & -1/2\,\overline{\mathbf{c}} \\
\ast & \ast & 0 & \ast & 0 & +1/2\,\overline{\mathbf{c}} \\
\ast & \overline{\mathbf{c}} & \ast & \ast & \ast & -1/2\,\overline{\mathbf{c}} \\
\ast & \ast & \ast & \ast & \ast & -1/2\,\overline{\mathbf{c}} \\
\ast & 0 & \ast & \ast & \ast & +1/2\,\overline{\mathbf{c}} \\
\hline
-1/2\,\overline{\mathbf{c}} & -1/2\,\overline{\mathbf{c}} & -1/2\,\overline{\mathbf{c}} & +1/2\,\overline{\mathbf{c}} & -1/2\,\overline{\mathbf{c}} &
\end{array}$$

cost with two lines is changed by $+1/2\,\overline{\mathbf{c}} + 1/2\,\overline{\mathbf{c}}$ *i.e.,* $+\overline{\mathbf{c}}$

$\ast$ = *nonzero cost*

Therefore, step 4 redistributes the zeroes without changing the optimal assignment.



cost with only one line is changed by $\frac{1}{2}\bar{c} - \frac{1}{2}\bar{c}$ *i.e., zero*

cost with no lines is changed by $-\frac{1}{2}\bar{c} - \frac{1}{2}\bar{c}$ *i.e.,* $-\bar{c}$

cost with two lines is changed by $+\frac{1}{2}\bar{c} + \frac{1}{2}\bar{c}$ *i.e.,* $+\bar{c}$

$$\begin{array}{ccccc|c}
\ast & \ast & \ast & 0 & \ast & -\frac{1}{2}\bar{c} \\
\ast & \ast & 0 & \ast & 0 & +\frac{1}{2}\bar{c} \\
\ast & \bar{c} & \ast & \ast & \ast & -\frac{1}{2}\bar{c} \\
\ast & \ast & \ast & \ast & \ast & -\frac{1}{2}\bar{c} \\
\ast & 0 & \ast & \ast & \ast & +\frac{1}{2}\bar{c} \\
\hline
-\frac{1}{2}\bar{c} & -\frac{1}{2}\bar{c} & -\frac{1}{2}\bar{c} & +\frac{1}{2}\bar{c} & -\frac{1}{2}\bar{c} &
\end{array}$$

$\ast$ = *nonzero cost*

Row reduction

$$\begin{array}{cc|cccc|} & & & \text{job} & & \\ & & 1 & 2 & 3 & 4 \\ \hline & A & 6 & 4 & 5 & 5 \\ \text{machine} & B & 7 & 4 & 5 & 6 \\ & C & 4 & 7 & 6 & 4 \\ & D & 5 & 3 & 4 & 7 \\ \hline \end{array} \Rightarrow \begin{array}{cc|cccc|} & & & \text{job} & & \\ & & 1 & 2 & 3 & 4 \\ \hline & A & 2 & 0 & 1 & 1 \\ \text{machine} & B & 3 & 0 & 1 & 2 \\ & C & 0 & 3 & 2 & 0 \\ & D & 2 & 0 & 1 & 4 \\ \hline \end{array}$$

Let's modify the original example somewhat, and repeat the row and column reductions.

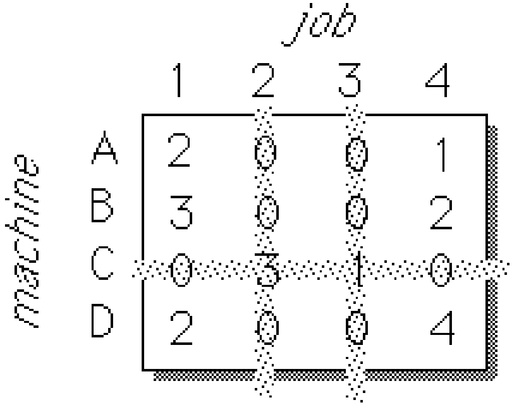
Column reduction

$$\begin{array}{cc|cccc} & & \multicolumn{4}{c}{\text{job}} \\ & & 1 & 2 & 3 & 4 \\ \hline & A & 2 & 0 & 1 & 1 \\ \text{machine} & B & 3 & 0 & 1 & 2 \\ & C & 0 & 3 & 2 & 0 \\ & D & 2 & 0 & 1 & 4 \end{array} \Rightarrow \begin{array}{cc|cccc} & & \multicolumn{4}{c}{\text{job}} \\ & & 1 & 2 & 3 & 4 \\ \hline & A & 2 & 0 & 0 & 1 \\ \text{machine} & B & 3 & 0 & 0 & 2 \\ & C & 0 & 3 & 1 & 0 \\ & D & 2 & 0 & 0 & 4 \end{array}$$

As we saw earlier, there is no zero-cost assignment possible with this matrix.

This can be determined by the fact that the zeroes can be covered with only 3 lines:

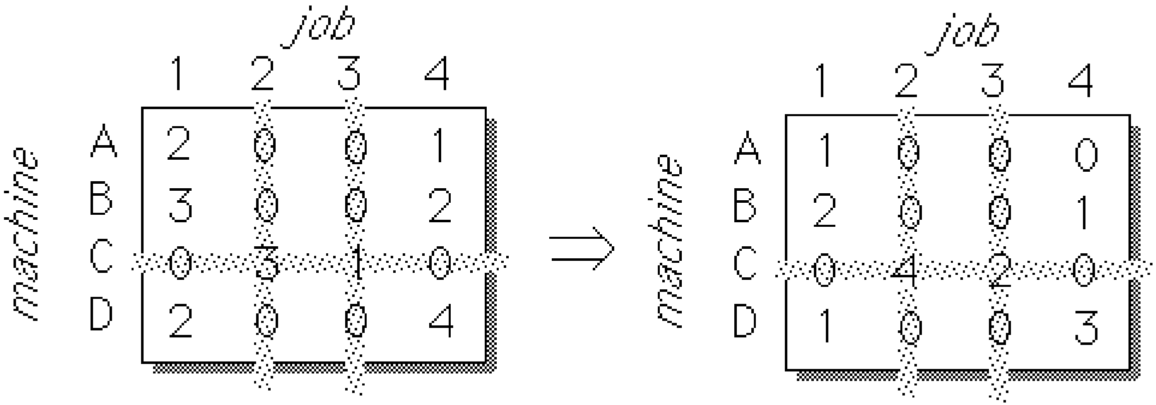
$$\begin{array}{cc|cccc|} & & & & \textit{job} & \\ & & 1 & 2 & 3 & 4 \\ \hline & A & 2 & 0 & 0 & 1 \\ \textit{machine} & B & 3 & 0 & 0 & 2 \\ & C & 0 & 3 & 1 & 0 \\ & D & 2 & 0 & 0 & 4 \\ \hline \end{array}$$



$$\begin{array}{cc|cccc|} & & & & \textit{job} & \\ & & 1 & 2 & 3 & 4 \\ \hline & A & 2 & 0 & 0 & 1 \\ \textit{machine} & B & 3 & 0 & 0 & 2 \\ & C & 0 & 3 & 1 & 0 \\ & D & 2 & 0 & 0 & 4 \\ \hline \end{array}$$

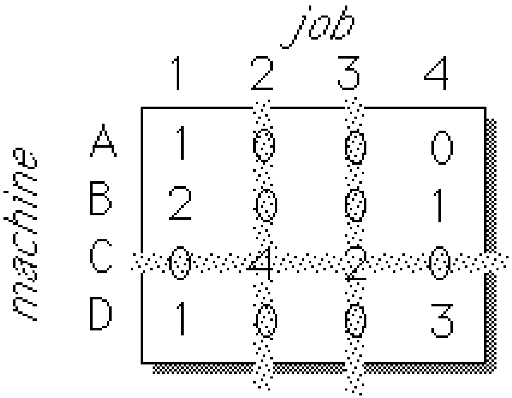
We therefore perform the reduction in step 4:

Step 4 locate the smallest unlined cost, $\bar{c}$. Subtract this cost from all unlined costs, and add to costs at intersections of lines.



$$\text{machine}\quad\begin{array}{c|cccc} & \multicolumn{4}{c}{\text{job}} \\ & 1 & 2 & 3 & 4 \\ \hline A & 2 & 0 & 0 & 1 \\ B & 3 & 0 & 0 & 2 \\ C & 0 & 3 & 1 & 0 \\ D & 2 & 0 & 0 & 4 \end{array} \quad\Longrightarrow\quad \text{machine}\quad\begin{array}{c|cccc} & \multicolumn{4}{c}{\text{job}} \\ & 1 & 2 & 3 & 4 \\ \hline A & 1 & 0 & 0 & 0 \\ B & 2 & 0 & 0 & 1 \\ C & 0 & 4 & 2 & 0 \\ D & 1 & 0 & 0 & 3 \end{array}$$

The new cost matrix has a zero not covered by a line:



$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline A & 1 & 0 & 0 & 0 \\ B & 2 & 0 & 0 & 1 \\ C & 0 & 4 & 2 & 0 \\ D & 1 & 0 & 0 & 3 \end{array}$$

The zeroes now require 4 lines in order to cover all of them!

job

machine

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline A & 1 & 0 & 0 & 0 \\ B & 2 & 0 & 0 & 1 \\ C & 0 & 4 & 2 & 0 \\ D & 1 & 0 & 0 & 3 \end{array}$$

In fact, there are two different zero-cost assignments, both of them optimal for this problem:

$$\text{machine}\;\begin{array}{c|cccc} & \multicolumn{4}{c}{\textit{job}} \\ & 1 & 2 & 3 & 4 \\ \hline A & 1 & 0 & 0 & \boxed{0} \\ B & 2 & \boxed{0} & 0 & 1 \\ C & \boxed{0} & 4 & 2 & 0 \\ D & 1 & 0 & \boxed{0} & 3 \end{array} \qquad \text{machine}\;\begin{array}{c|cccc} & \multicolumn{4}{c}{\textit{job}} \\ & 1 & 2 & 3 & 4 \\ \hline A & 1 & 0 & 0 & \boxed{0} \\ B & 2 & 0 & \boxed{0} & 1 \\ C & \boxed{0} & 4 & 2 & 0 \\ D & 1 & \boxed{0} & 0 & 3 \end{array}$$