An Algorithm for the Assignment Problem

In Sec. 8.3, we pointed out that the transportation simplex method can be used to solve assignment problems but that a *specialized* algorithm designed for such problems should be more efficient. We now will describe a basic algorithm (sometimes called the **Hungarian algorithm**) of this type. We will focus just on the key ideas without filling in all the details needed for a complete computer implementation.

The algorithm operates directly on the *cost table* for the problem. More precisely, it converts the original cost table into a series of *equivalent* cost tables until it reaches one where an optimal solution is obvious. This final equivalent cost table is one consisting of only *positive* or *zero* elements where all the assignments can be made to the zero element positions. Since the total cost cannot be negative, this set of assignments with a zero total cost is clearly optimal. The question remaining is how to convert the original cost table into this form.

The key to this conversion is the fact that one can add or subtract any constant from every element of a row or column of the cost table without really changing the problem. That is, an optimal solution for the new cost table must also be optimal for the old one, and conversely.

To illustrate these ideas, consider the cost table for the Job Shop Co. problem given in Table 8.25. To convert this cost table into an equivalent cost table, suppose that we subtract 11 from every element in row 1, which yields:

	1	2	3	4
1	2	5	1	0
2	15	M	13	20
3	5	7	10	6
4(D)	0	0	0	0

Since any feasible solution must have exactly one assignment in row 1, the total cost for the new table must always be exactly 11 less than for the old table. Hence, the solution which minimizes total cost for one table must also minimize total cost for the other.

Notice that, whereas the original cost table had only strictly positive elements in the first three rows, the new table has a zero element in row 1. Since the objective is to obtain enough strategically located zero elements to yield a complete set of assignments, this process should be continued on the other rows and columns. Negative elements are to be avoided, so the constant to be subtracted should be the minimum element in the row or column. Doing this for rows 2 and 3 yields the following equivalent cost table:

	1	2	3	4
1	2	5	1	[0]
2	2	M	[0]	7
3	[0]	2	5	1
4(D)	0	[0]	0	0

This cost table has all the zero elements required for a complete set of assignments, as shown by the four boxes, so these four assignments constitute an *optimal solution* (as claimed earlier for this problem). The total cost for this optimal solution is seen in Table 8.25 to be $Z = 29$, which is just the sum of amounts that have been subtracted from rows 1, 2, and 3.

Unfortunately, an optimal solution is not always obtained quite so easily, as we now illustrate with the assignment problem formulation of Option 2 for the Better Products Co. problem shown in Table 8.29.

Because this problem's cost table already has zero elements in every row but the last one, let us begin the process of converting to equivalent cost tables by subtracting the minimum element in each column from every entry in that column. The result is shown below.

	1	2	3	4	5(D)
1a	80	0	30	120	0
1b	80	0	30	120	0
2a	60	60	M	80	0
2b	60	60	M	80	0
3	0	90	0	0	M

Now *every* row and column has at least one zero element, but a complete set of assignments with zero elements is *not* possible this time. In fact, the maximum number of assignments that can be made in zero element positions is only 3. (Try it.) Therefore, one more idea must be implemented to finish solving this problem that was not needed for the first example.

This idea involves a new way of creating *additional* positions with zero elements without creating any negative elements. Rather than subtracting a constant from a *single* row or column, we now add or subtract a constant from a *combination* of rows and columns.

This procedure begins by drawing a set of lines through some of the rows and columns in such a way as to *cover all the zeros*. This is preferably done with a *minimum* number of lines, as shown in the next cost table.

	1	2	3	4	5(D)
1a	80	0	30	120	0
1b	80	0	30	120	0
2a	60	60	M	80	0
2b	60	60	M	80	0
3	0	90	0	0	M

Notice that the minimum element not crossed out is 30 in the two top positions in column 3. Therefore, subtracting 30 from every element in the entire table, i.e., from every row or from every column, will create a new zero element in these two positions. Then, in order to restore the previous zero elements and eliminate negative elements, add 30 to each row or column with a line covering it—row 3 and columns 2 and 5(*D*). The result is given below.

	1	2	3	4	5(D)
1a	50	0	0	90	0
1b	50	0	0	90	0
2a	30	60	M	50	0
2b	30	60	M	50	0
3	0	120	0	0	M

A shortcut for obtaining this cost table from the preceding one is to subtract 30 from just the elements without a line through them and then add 30 to every element that lies at the intersection of two lines.

With this new cost table, it now is possible to make four assignments to zero element positions, but still not five. (Try it.) Therefore, we repeat the above procedure, where four lines (the same number as the maximum number of assignments) now are the minimum needed to cover all zeroes. One way of doing this is shown below.

	1	2	3	4	5(D)
1a	50	0	0	90	0
1b	50	0	0	90	0
2a	30	60	M	50	0
2b	30	60	M	50	0
3	0	120	0	0	M

The minimum element not covered by a line is again 30, but now in the first position in rows 2*a* and 2*b*. Therefore, we subtract 30 from every *uncovered* element and add 30 to every *doubly covered* element (except for ignoring elements of *M*), which gives the following equivalent cost table.

	1	2	3	4	5(D)
1a	50	[0]	0	90	30
1b	50	0	[0]	90	30
2a	[0]	30	M	20	0
2b	0	30	M	20	[0]
3	0	120	0	[0]	M

This table actually has several ways of making a complete set of assignments to zero element positions (several optimal solutions), including the one shown by the five boxes. The resulting total cost is seen in Table 8.29 to be

$$Z = 810 + 840 + 800 + 0 + 840 = 3,290.$$

PROBLEMS

8S-1. Reconsider the assignment problem presented in Prob. 8.3-2. Manually apply the Hungarian algorithm to solve this problem.

8S-2. Reconsider Prob. 8.3-4. See its formulation as an assignment problem in the back of the book. Manually apply the Hungarian algorithm to solve this problem.

8S-3. Reconsider the assignment problem formulation of Option 2 for the Better Products problem presented in Table 8.29. Suppose that the cost of having plant 1 produce product 1 is reduced from 820 to 720. Solve this problem by manually applying the Hungarian algorithm.

8S-4. Manually apply the Hungarian algorithm to solve the assignment problem having the following cost table:

		Job		
		1	2	3
	1	M	8	7
Person	2	7	6	4
	3(D)	0	0	0

8S-5. Manually apply the Hungarian algorithm to solve the assignment problem having the following cost table:

		Task			
		1	2	3	4
Assignee	*A*	4	1	0	1
	B	1	3	4	0
	C	3	2	1	3
	D	2	2	3	0

8S-6. Manually apply the Hungarian algorithm to solve the assignment problem having the following cost table:

		Task			
		1	2	3	4
Assignee	*A*	4	6	5	5
	B	7	4	5	6
	C	4	7	6	4
	D	5	3	4	7