

# Linear Programming Notes VIII: The Transportation Problem

## 1 Introduction

Several examples during the quarter came with stories in which the variables described quantities that came in discrete units. It makes sense that you can produce coffins in only whole number units. It is hard to imagine selling  $\frac{2}{3}$  of a chair or  $\frac{1}{2}$  of a table. So far we have ignored these constraints. In applications, one must take integer constraints seriously. An intelligent, but naive, way to deal with the constraints is to solve the problem assuming that the constraints are not present and then round your solution to the nearest integer value. In many situations this technique is not only sensible, but gives good answers. If you round down the number of items you produce in a production problem, then you are likely to maintain feasibility and you may arrive at the true solution to the problem. In general, the method won't work. Rounding (even rounding down) may destroy feasibility or the true solution may not be close to the solution of the problem solved without imposing integer constraints. The general topic of Integer Programming deals with the problem directly. The theory of Integer Programming (or Linear Integer Programming) is not as complete as the theory of Linear Programming. Integer Programming problems are more difficult to solve than LPs. Econ 172B describes some general approaches.

In this section I introduce problems that have a special property. In these problems, it is especially natural to impose the constraint that the variables take on integer values. Hence the problems are, strictly speaking, not linear programming problems. Nevertheless, aside from the integer constraint, the problems are linear. Moreover, the problems are so special that when you solve them as LPs, the solutions you get automatically satisfy the integer constraint. (More precisely, if the data of the problem is integral, then the solution to the associated LP will be integral as well.)

## 2 The Transportation Problem

### 2.1 Formulation

The Transportation Problem was one of the original applications of linear programming models. The story goes like this. A firm produces goods at  $m$  different supply centers. Label these  $i = 1, \dots, m$ . The supply produced at supply center  $i$  is  $S_i$ . The demand for the good is spread out at  $n$  different demand centers. Label these  $j = 1, \dots, n$ . The demand at the  $j$ th demand center is  $D_j$ . The problem of the firm is to get goods from supply centers to demand centers at minimum cost. Assume that the cost of shipping one unit from supply center  $i$  to demand center  $j$  is  $c_{ij}$  and that shipping cost is linear. That means that

if you shipped  $x_{ij}$  units from supply center  $i$  to demand center  $j$ , then the cost would be  $c_{ij}x_{ij}$ .

I have already done one of the steps of formulating the problem: I have introduced variables. Let me be explicit. Define  $x_{ij}$  to be the number of units shipped from supply center  $i$  to demand center  $j$ . The problem is to identify the minimum cost shipping schedule. The constraints are that you must (at least) meet demand at each demand center and cannot exceed supply at each supply center.

The cost of the schedule, by the linearity assumption, is given by

$$\min \sum_{i=1}^m \sum_{j=1}^n x_{ij}c_{ij}.$$

Now let's figure out the constraints. Consider supply center  $i$ . The total amount shipped out of supply center  $i$  is  $\sum_{j=1}^n x_{ij}$ . Think about this expression.  $x_{ij}$  is what you ship from  $i$  to  $j$ . From  $i$  you can ship to any demand center ( $j = 1, \dots, n$ ). The sum above just adds up the total shipment from supply center  $i$ . This quantity cannot exceed the supply available. Hence we have the constraint

$$\sum_{j=1}^n x_{ij} \leq S_i \text{ for all } i = 1, \dots, m.$$

Similarly, the constraints that guarantee that you meet the demand at each of the demand centers look like:

$$\sum_{i=1}^m x_{ij} \geq D_j \text{ for all } j = 1, \dots, n.$$

Consider the feasibility of the problem. The only way that the problem can be feasible is if total supply exceeds total demand ( $\sum_{j=1}^n D_j \leq \sum_{i=1}^m S_i$ ). If this equation did not hold, then there would be excess demand. There would be no way to meet all of the demand with available supply. If there is enough supply, then you should be able to convince yourself that you can satisfy the constraints of the problem. That is, the problem is feasible unless there is excess demand. It is conventional to assume that the total supply is equal to the total demand. If so, that is, if

$$\sum_{j=1}^n D_j = \sum_{i=1}^m S_i,$$

then all of the constraints in the problem must hold as equations (that is, when total supply equals total demand, then a feasible transportation plan exactly meets demand at each demand center and uses up all of the supply at each supply center). (In cases where there is excess supply, you can transform the problem into one in which supply is equal to demand by assuming that you can freely dispose of the extra supply.)

After making the simplification that total supply equals total demand, we arrive at the standard formulation of the transportation problem. The problem

provides  $m$  supplies  $S_i$  for  $i = 1, \dots, m$ ,  $n$  demands  $D_j$  for  $j = 1, \dots, n$  that satisfy  $\sum_{j=1}^n D_j = \sum_{i=1}^m S_i$ , and costs  $c_{ij}$ . The objective is to find a transportation plan denoted by  $x_{ij}$  to solve:

$$\min \sum_{i=1}^m \sum_{j=1}^n x_{ij} c_{ij}$$

subject to

$$\sum_{j=1}^n x_{ij} = S_i \text{ for all } i = 1, \dots, m.$$

and

$$\sum_{i=1}^m x_{ij} = D_j \text{ for all } j = 1, \dots, n.$$

In this problem it is natural to assume that the variables  $x_{ij}$  take on integer values (and non-negative ones). That is, you can only ship items in whole number batches.

## 2.2 Discussion

The transportation problem is an optimization problem with a linear objective function and linear constraints. If we ignore the restriction that the variables take on integer values, then it would fall into our standard framework. We can solve the transportation problem using Excel.

The transportation problem has a lot of special structure. For example, each variable appears in exactly two constraints (with a non-zero coefficient). When a variable has a non-zero coefficient, the coefficient is either plus or minus 1. Because of this special structure, two things turn out to be true. First, there are alternative methods of solving transportation problems that are more efficient than the standard simplex algorithm. This turns out to be important in practice, because real-world transportation problems have enormous numbers of variables. Second, because of the special structure, it is possible to solve the transportation problem in whole numbers. That is, if the data of the problem (supplies, demands, and costs) are all whole numbers, then there is a whole number solution. The significance of this property is that you do not need to impose the difficult to handle integer constraints in order to get a solution that satisfies the constraints.

I will not explain completely why you can always find integer solutions to transportation problems. Several things are worth noting. It is not true in general. For example, the recurring example of the course (the problem that we used to illustrate the simplex algorithm) started with whole number data, but its solution involved fractions.

There are two intuitions about why transportation problems have integer solutions. One intuition is that corners of the feasible sets of transportation problems must have whole number coordinates. That is, if you solve a subset

of  $k$  constraints using only  $k$  variables, the solution will involve whole numbers. This intuition is geometric. You know that solutions to LPs arise at corners. If you can see that corners of the feasible set have whole number coordinates, you are in business. (Note: I have just claimed that this property holds. I have not proved it. The proof is complicated, so you have a right to be skeptical about the claim.)

The other intuition is algebraic. In the simplex algorithm, you get fractions because you must divide by the element you pivot on. In the transportation problem pivot elements will always be 1, so there is no need to divide.

### 2.3 Short History

People thought the transportation problem up early in the Second World War. It was used to determine how to move troops (located, for example, at training bases in different parts of the United States) to battlegrounds in Europe and Asia.

### 2.4 The Dual of the Transportation Problem

Every LP has a dual. The neatest way to write the dual of the transportation problem is: Find  $u_i$  for  $i = 1, \dots, m$  and  $v_j$  for  $j = 1, \dots, n$  to solve:

$$\max - \sum_{i=1}^m u_i S_i + \sum_{j=1}^n v_j D_j$$

subject to

$$u_i - v_j \leq c_{ij} \text{ for all } i = 1, \dots, m \text{ and } j = 1, \dots, n.$$

This paragraph gives an explanation of how I arrived at the dual. It is only an outline. I will not ask you to write the dual of a problem as complicated as the transportation problem. So the material in this paragraph is optional. On the other hand, it is extremely useful to know where duals come from. Arriving at the formulation of the dual takes a bit of care. The tedious method is to transform the original transportation problem so that it is in standard form, take the dual of that, and simplify. The clever method is to notice that transportation problem was written as a minimization (so that the dual will be a maximization); it had equality constraints (so that the dual variables will be unconstrained); its variables were constrained (so that in the dual the constraints are inequalities). I did one other tricky thing. I let  $-u_i$  be the name of the variable for the  $i$ th constraint in the transportation problem. This created negative signs in the objective function and constraints in the dual. This definition is mathematically irrelevant (since the variable is unconstrained in sign), but leads to a form that is consistent with the story I tell in the next paragraph.

Let me now try to interpret the dual. Here is a way to think of the possibility. In the original transportation problem, the seller faces the problem of getting goods from the supply centers to the demand centers. The only way to do this

is by using conventional shipping lines and paying costs described by  $c_{ij}$ . Now, for the purpose of the dual, imagine that someone offers to transport the goods for the supplier. This mysterious shipper offers to buy goods from the supplier at each supply center (at the price  $u_i$  at supply center  $i$ ) and resell them at demand center  $j$  at the price  $v_j$ . Somehow the mysterious shipper manages to get the goods where they belong. The original seller doesn't care how the goods get where they should be, as long as shipping cost is not too great.  $c_{ij}$  is the amount it would cost to move an item from supply center  $i$  to demand center  $j$  using conventional methods. Using the mysterious shipper it would cost  $v_j - u_i$  (because the seller must pay  $v_j$  to get the good back, but receives  $u_i$  when he sells it. Therefore, if the constraints in the dual are satisfied, then it is no more expensive to use the shipper than to use conventional shipping methods. The dual objective function is the amount that the mystery shipper earns by buying all of the supply and then reselling it at demand centers. This discussion leads to the interpretation of the dual. The mystery shipper sets prices at each supply and demand location so that the net cost of shipping an item is no greater than the direct ( $c_{ij}$ ) cost, and does so to maximize net revenue.

## 2.5 Example

Here is an example that is inspired by a similar problem in Hillier and Lieberman.

A lumber company has three sources of wood and five markets where wood is demanded. The annual quantity of wood available in the three sources of supply are 15, 20, and 15 million board feet respectively. The amount that can be sold at the five markets is 11, 12, 9, 10, and 8 million board feet, respectively. The company currently transports all of the wood by train. It wishes to evaluate its transportation schedule, possibly shifting some or all of its transportation to ships. The unit cost of shipment (in \$10,000 along the various routes using both methods is described in the table below.

Supply	Market 1	Market 2	Market 3	Market 4	Market 5
<i>A</i>	51	62	35	45	56
<i>B</i>	59	68	50	39	46
<i>C</i>	49	56	53	51	37

Cost per Unit of Rail Transport

Supply	Market 1	Market 2	Market 3	Market 4	Market 5
<i>A</i>	48	68	48	none	54
<i>B</i>	66	75	55	49	57
<i>C</i>	none	61	64	59	50

Cost per Unit of Ship Transport

The management needs to decide to what extent to continue to rely on rail transportation. Evaluate the following options and make a recommendation about what to do.

1. How much does it cost use rail transport exclusively?
2. How much does it cost to use ships exclusively?
3. How much does it cost to use the cheapest available mode of transportation on each route?
4. Suppose that there is an annual cost of \$100,000 to operate any ships (but that this cost does not vary with the number of shipping lines kept open). What is the optimal transportation plan?
5. How would your answer change if you learned that the supply at Center *B* and the demand at market 3 were both expected to increase by 10 million board feet?

I wrote a spreadsheet that describes the problem. It is available. On the spreadsheet I wrote three cost arrays. One represents the costs of train transportation; the second of ship transportation; and the third the minimum (route by route). For the routes that for which shipping was not feasible (the “none” entries in the cost table), I substituted a large number.

I first solved the problem using rail transportation. I obtained

Supply	Market 1	Market 2	Market 3	Market 4	Market 5
<i>A</i>	6	0	9	0	0
<i>B</i>	2	0	0	10	8
<i>C</i>	3	12	0	0	0

#### Solution Using only Rail Transit

The cost of this transportation plan is 2316. (I also computed the cost of this transit schedule if ships were used instead of trains, this cost is 5530; 2298 is the cost of this shipping plan using the minimum cost method on each route. There is no a priori reason why the cost of the shipping cost should be more than the rail cost. The min cost must be lower (or equal). The fact that it is strictly lower means that a positive amount of the lumber was transported over routes that are less expensive to use ships than trains.

Next I solved the problem using only ship transit. (I just copied the original spreadsheet and changed the objective function from train value to ship value.) This is the solution.

Supply	Market 1	Market 2	Market 3	Market 4	Market 5
<i>A</i>	11	0	4	0	0
<i>B</i>	0	0	5	10	5
<i>C</i>	0	12	0	0	3

#### Solution Using only Ship Transit

The cost of this plan (using ships) is 2654. It would be 2354 using trains and 2321 using the minimum cost method. Note that even though the transportation

plan is optimal for ships, it costs more to use ships than trains (but if you were going to use trains it would be cheaper still to use the first transportation plan). Finally, as before, if you can use the minimum cost method, you would have an even lower cost.

At this point I have answered the first question (2316) and the second question (2652). If you must use only one mode of transportation, it does not pay to switch. We can also conclude that using shipping selectively is profitable. We do not know how profitable until we solve the third problem. I did and obtained:

Supply	Market 1	Market 2	Market 3	Market 4	Market 5
<i>A</i>	6	0	9	0	0
<i>B</i>	2	0	0	10	8
<i>C</i>	3	12	2	0	0

#### Solution Using Min Cost Transit

The cost of this transportation plan is 2298 (2316 if all were transported by train and 5530 if all were transported by ship). This schedule is identical to the first one. The solution is less expensive than using only ships or only trains. In fact, we can conclude that it is worth \$180,000 (remember units are \$10,000) per year to have the option to use ships. Provided that the cost of having a ships is less than \$180,000 it is worth operating the minimum cost plan. Ships are used for only one route: connecting *A* to 1.

The last question asks you to redo the problem under the assumption that the supply at *B* is 30 (instead of 20) and the demand at 3 is 19 (instead of 9). I re-solved the problem and obtained the cost of 2774 for only trains using these routes:

Supply	Market 1	Market 2	Market 3	Market 4	Market 5
<i>A</i>	0	0	15	0	0
<i>B</i>	8	0	4	10	8
<i>C</i>	3	12	0	0	0

#### Solution 2 Using only Rail Transit

The basis changed (it is no longer profitable to use the *A* to 1 route). Using only ships the solution becomes:

Supply	Market 1	Market 2	Market 3	Market 4	Market 5
<i>A</i>	11	0	4	0	0
<i>B</i>	0	0	15	10	5
<i>C</i>	0	12	0	0	3

#### Solution 2 Using only Ship Transit

with cost equal to 3202. The only difference between this transportation plan and the original solution to the problem using ships is that now ten extra

units are shipped from  $B$  to 3. The cost went up by 550. On the other hand, going from the first to the second train made the cost go up by less ( $458 = 2774 - 2316$ ) than 550. That is, although the direct transportation cost from  $C$  to 3 is 55 per unit, by using other routes (at least for some of the units) the extra demand can be transported at a smaller price.

Finally, when you solve the min problem you again find that the solution agrees with the solution from the train transportation problem. Now, however, you don't use any ships. That is, the additional demand makes it optimal to only use rail transportation.

### 3 The Assignment Problem

#### 3.1 Introduction

The Assignment Problem is a special case of the transportation problem in which there are equal numbers of supply and demand centers, and that all demands and supplies are equal to one. Sometimes you interpret the "costs" ( $c_{ij}$ ) as benefits, and solve a maximization problem instead of a minimization problem. This change of interpretation has adds no theoretical problems.

The Assignment Problem deserves special attention because it is an interesting special case. The usual story that comes with it goes like this. You are the manager of a little league baseball team. After carefully watching the nine children on your team, you can assign the value of having player  $i$  play position  $j$ . (I am assuming that there are nine positions on a baseball team. This is still true in the National League.) Denote this value  $a_{ij}$ . The objective is to find an assignment - that is a position for each player on the team - such that each player plays only one position and each position has only one player (this is, there is only one pitcher and even the best player can play only one position) that maximizes the total possible value. If we let  $x_{ij}$  be equal to 1 if player  $i$  is assigned to position  $j$  and equal to zero otherwise, then the problem is to find  $x_{ij}$  to solve:

$$\max \sum_{i=1}^n \sum_{j=1}^n x_{ij} a_{ij}$$

subject to

$$\sum_{i=1}^n x_{ij} = 1 \text{ for } j = 1, \dots, n$$

and

$$\sum_{j=1}^n x_{ij} = 1 \text{ for } i = 1, \dots, n.$$

Also, the variables  $x_{ij}$  must take on the values 0 or 1 (otherwise your assignment would involve cutting people into pieces. This is very messy and usually does not improve the performance of the baseball team.)



The assignment model has a wide range of applications. You can imagine matching women to men; workers to jobs; and so on. Variations of the model are used to assign medical residents to hospital training programs. Complicated versions of the model are used for scheduling (classes to classrooms or teams in professional sports leagues).

### 3.2 The Hungarian Method

The assignment problem is a linear programming problem (with the additional constraint that the variables take on the values zero and one). In general, the additional constraint makes the problem quite difficult. However, like the transportation problem, the assignment problem has the property that when you solve the problem ignoring the integer constraints you still get integer solutions. This means that the simplex algorithm solves assignment problems. Assignment problems have so much special structure that there are simpler algorithms available for solving them. In this section, I will describe one of the algorithms, called the Hungarian method. I suspect that it is politically incorrect now to name a method after a country. I believe (but I did not verify) that the name is a tribute to the Hungarian mathematicians that originally discovered the algorithm.

I will illustrate the algorithm with an example. Consider the assignment problem with the costs given in the array below.

	1	2	3	4
<i>A</i>	10	7	8	2
<i>B</i>	1	5	6	3
<i>C</i>	2	10	3	9
<i>D</i>	4	3	2	3

This array describes an assignment problem with four people (labeled *A*, *B*, *C*, and *D*) and four jobs (1, 2, 3, 4). The first person has a cost 10 if assigned to the first job; a cost 7 if assigned to the second job; etc. The goal is to assign people to jobs in a way that minimizes total cost.

The algorithm uses a simple observation and one trick. The observation is that you can subtract a constant from any row or column without changing the solution to the problem. Take the first row (the costs associated with *A*). All of these numbers are at least two. Since you must assign person *A* to some job, you must pay at least two no matter what. If you'd like, think of that as a fixed cost and further costs as variable costs depending on the job assigned to the first person. Hence if I reduce all of the entries in the first row by two, I do not change the optimal assignment (I lower the total cost by two). Doing so leaves this table:

	1	2	3	4
<i>A</i>	8	5	6	0
<i>B</i>	1	5	6	3
<i>C</i>	2	10	3	9
<i>D</i>	4	3	2	3

Again, the solution to the problem described by the second table is exactly the same as the solution to the first problem. Continuing in this way I can subtract the “fixed cost” for the other three people (rows) so that there is guaranteed to be at least one zero in each row. I obtain:

	1	2	3	4
<i>A</i>	8	5	6	0
<i>B</i>	0	4	5	2
<i>C</i>	0	8	1	7
<i>D</i>	2	1	0	1

I’m not done using this observation yet. Just as I can subtract a constant from any row, I can subtract a constant from any column. Take the second column. It says that no matter who you assign to the second job, it will cost at least 1. Treat the 1 as a fixed cost and subtract it. Since it cannot be avoided it does not influence your solution (it does influence the value of the solution). Once you make this reduction you get:

	1	2	3	4
<i>A</i>	8	4	6	0
<i>B</i>	0	3	5	2
<i>C</i>	0	7	1	7
<i>D</i>	2	0	0	1

This is the end of what we can do with the simple observation. Now it is time to use the observation. The last table is simpler than the original one. It has the property that there is a zero in every row and in every column. All of the entries are non-negative. Since you want to find an assignment that minimizes total cost, it would be ideal if you could find an assignment that only pairs people to jobs when the associated cost is zero. Keep this in mind: The goal of the computation is to write the table in a way that is equivalent (has the same solution) as the original problem and has a zero-cost assignment. I have just finished the step in which you reduce the costs so that there is at least one zero in every row and every column. The example demonstrates that this is not enough.

If you think about the table, you will see that this is not possible. If you try to come up with a zero cost assignment, you must assign *A* to 4 (the only zero in the row for *A* is in the 4 column) and you must assign *B* to 1. However, the only way to get a zero cost from *C* is to assign it to 1 as well. I can’t do this, because I have already assigned *B* to 1. If you have followed up until now, you will be able to conclude that you should do the next best thing: assign *C* to job 3 (at the cost 1) and then *D* to 2. This yields the solution to the problem (*A* to 4; *B* to 1; *C* to 3; *D* to 2). It is not, however, an algorithm. We made the final assignments by guessing. (You should be sure that this is the solution. I argued that it is impossible to solve the problem at cost zero, but then demonstrated that it is possible to solve the problem at the next best cost, one.)

To turn the intuition into an algorithm, we need a trick. When I subtracted a constant from each row, I did so in order to make the smallest element of each row 0. What I would like to do is to continue to create new cheap assignments without changing the essence of the problem. The trick is to eliminate the zeroes in the table and then try to reduce the remaining values.

Here I repeat the past table:

	1	2	3	4
<i>A</i>	<u>8</u>	<u>4</u>	<u>6</u>	<u>0</u>
<i>B</i>	0	3	5	2
<i>C</i>	0	7	1	7
<i>D</i>	<u>2</u>	<u>0</u>	<u>0</u>	<u>1</u>

I have crossed out two rows and one column. Doing so “covers up” all of the zeros. Now look at the uncovered cells and find the smallest number (it turns out to be one). If I subtracted one from each cell in the entire matrix, then I would leave the basic problem unchanged (that is, I would not change the optimal assignment) and I would “create” a new low cost route (*C* to 3). That is the good news. The bad news is that some entries (covered by lines) would become negative. This is bad news because if there are negative entries, there is no guaranteed that a zero-cost assignment really minimizes cost. So reverse the process by adding the same constant you subtracted from every entry (1) to each row and column with a line through it. Doing so creates this cost matrix:

	1	2	3	4
<i>A</i>	9	4	6	0
<i>B</i>	0	2	4	1
<i>C</i>	0	6	0	6
<i>D</i>	3	0	0	1

The beauty of this table is that it again is non-negative. It turns out that using this matrix it is possible to make another minimum cost assignment. In fact, using this table, we can come up with an optimal assignment with cost zero. It agrees with our intuition (*A* to 4; *B* to 1; *C* to 3; *D* to 2). You can go back to the original matrix of costs to figure out what the total cost is:  $9 = 2 + 1 + 3 + 3$ . Mechanically:

1. Subtract the minimum number from each zero to leave one zero element in each row.
2. Subtract the minimum number from each column to leave one zero element in each column.
3. Find the minimum number of lines that cross out all of the zeroes in the table.
4. From all of the entries that are not crossed out, find the minimum number (it should be positive). If the minimum is zero, then you haven't crossed

out enough entries. If all of the entries are crossed out, then you already should be able to find a zero cost assignment.<sup>1</sup>

5. Subtract the number that you found in Step 4 from all of the entries that haven't been crossed out. Do not change the entry in any cell that has one line through it. Add the number to those entries that have two lines through it.
6. Return to Step 1.

The first two steps are simple. They make the problem more transparent. The third and fourth steps are general versions of the first two steps. What you do in these steps is redistribute the costs in a way that does not change the solution.

Step 3 is the mysterious step. I ask you to cross out all of the zeroes in the table using the minimum number of lines. I recommend that you do this by finding the row or column that has the most zeroes; cross that one out. Next, cross out the row or column that has the most remaining uncrossed zeroes. (There may be more than one way to do this.) Continue until you are done.

In Step 5 you do two things. First, you subtract the number you found in Step 4 from **every** element of the table. As you know, this does not change the solution. It does, however, create negative numbers. Hence you must do something to restore non-negativity in the cost table (otherwise you cannot apply the rule that you want to find a zero-cost assignment to solve the problem). You do this by adding the constant back to every row or column that you draw a line through. When all is done, you are left with a table that satisfies the properties in Step 5. All entries that are not "lined" go down; the ones that have one line through them stay the same (go down and then go up by the same amount); the ones that have two lines (none will have three) go up (they go down, but then they go up twice).

You are done when you reach a stage in which you can find a zero-cost assignment. I won't provide a general procedure for doing this. It is natural to start by looking to see if any row or column has exactly one zero in it. If it does, you must include the assignment corresponding to that cell. Do so, cross out the corresponding row and column, and solve the remaining (smaller) problem. If each row and column contains at least two zeroes, make one assignment using an arbitrary row and column (with a zero cell) and continue. The problems that I ask you to solve will be small enough to solve by trial and error.

There is one other loose end. I have not demonstrated that the algorithm must give you a solution in a finite number of steps. The basic idea is that each step lowers the cost of your assignment. Verifying this requires a small argument. I will spare you.

Here is another example.

---

<sup>1</sup>You are done if you need to draw as many lines as there are rows or columns in the cost table.

	1	2	3	4	5
<i>A</i>	81	14	36	40	31
<i>B</i>	20	31	25	26	81
<i>C</i>	30	87	19	70	65
<i>D</i>	23	56	60	18	45
<i>E</i>	12	15	18	21	100

I will first subtract the minimum element in each row:

	1	2	3	4	5
<i>A</i>	67	0	22	26	17
<i>B</i>	0	11	5	6	61
<i>C</i>	11	68	0	51	46
<i>D</i>	5	38	42	0	27
<i>E</i>	0	3	6	9	88

Next, I subtract the minimum element from each column (only the fifth column has no zero in it).

	1	2	3	4	5
<i>A</i>	<u>67</u>	<u>0</u>	<u>22</u>	<u>26</u>	<u>0</u>
<i>B</i>	0	11	5	6	44
<i>C</i>	11	68	0	51	29
<i>D</i>	5	38	42	0	10
<i>E</i>	0	3	6	9	71

This array does not permit a zero-cost solution (both 2 and 5 must be matched with *A*). Hence we need to change it.

	1	2	3	4	5
<i>A</i>	70	0	25	29	0
<i>B</i>	0	8	5	6	41
<i>C</i>	11	65	0	51	26
<i>D</i>	5	35	42	0	7
<i>E</i>	0	0	6	9	67

From this array we can find a zero-cost assignment. The solution is *A* to 5; *B* to 1; *C* to 3; *D* to 4; and *E* to 2. Using the costs from the original table, the cost of this plan is:

$$31 + 20 + 19 + 18 + 15 = 103.$$