CRITICAL STUDY ON APPLLICATION OF LINEAR PROGRAMMING MODELS IN TRANSPORTATION PROBLEM

Jayashree Sinha [1] , Dr. Niranjan Kumar Mishra [2]

[1] Research Scholar, Department of Mathematics, RKDF University Ranchi

[2] Assistant Professor, Department of Mathematics, RKDF University Ranchi

Received : 05.10.2020 Accepted : 04.11.2020 Published : 05:12.2020

ABSTRACT :

Transportation problem is a special kind of Linear Programming Problem (LPP) in which goods are transported from a set of sources to a set of destinations subject to the supply and demand of the sources and destination respectively such that the total cost of transportation is minimized. Minimizing the cost of transporting products from production and storage locations to demand centers is an essential part of maintaining profitability for companies who deal with product distribution. Since transportation costs are generally not controllable, minimizing total cost requires making the best product routing decisions. This essential problem was first formulated as a linear programming problem in the early 1940's and is popularly known as the transportation problem. This Paper reflects Critical Study on application of Linear Programming Models in transportation problem.

Keywords : Linear Programming, transportation, supply, shipping,

I. INTRODUCTION :

Transportation problem is a special kind of Linear Programming Problem (LPP) in which goods are transported from a set of sources to a set of destinations subject to the supply and demand of the sources and destination respectively such that the total cost of transportation is minimized. It is also sometimes called as Hitchcock problem.

Types of Transportation problems:

Balanced: When both supplies and demands are equal then the problem is said to be a balanced transportation problem.

Unbalanced: When the supply and demand are not equal then it is said to be an unbalanced transportation problem. In this type of problem, either a dummy row or a dummy column is added according to the requirement to make it a balanced problem. Then it can be solved similar to the balanced problem.

Hence improvement in the transportation plan has significant result to a company's bottom-line. Research has reflected that a five percent reduction in transportation cost has similar impact as around 30 percent increase in sales. Means of sustain cost savings. Additionally, improvement in transportation usually provide better service levels to customers.

II. LINEAR PROGRAMMING IN TRANSPORTATION PROBLEM :

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 $i = 1, \ldots, m$. 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 $j = 1, \ldots, n$. The demand at the jth demand center is Dj. 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.

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" (cy) 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 our 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 ay. 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 xy be equal to 1 if player i is assigned to position j and equal to zero otherwise, then the problem is to find Xij to solve:

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

Subject to $\sum_{i=1}^{n} x_{ij} = 1 \ for \ j = 1, \ldots\ldots, n$

And $\sum_{j=1}^{n} x_{ij} = 1 \ for \ j = 1, \ldots\ldots, n$

Also, the variables x_{ij} must take on the values 0 or 1 (otherwise our 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. We can imagine matching women to men; workers to jobs; and so on. Variations of the model are used to assign medical residents

ISSN- 2394-5125 VOL 7, ISSUE 12, 2020

to hospital training programs. Complicated versions of the model are used for scheduling (classes to classrooms or teams in professional sports leagues).

III. 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 we 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 illustrate the algorithm with an example. Consider the assignment problem with the costs given in the array below.

	1	2	3	4
A	1	7	8	2
B	1	5	6	3
C	2	1	3	9
D	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 we 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 we must assign person A to some job, we must pay at least two no matter what. If we'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
A	8	5	6	0
B	1	5	6	3
C	2	1	3	9
D	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
A	8	5	6	0
B	0	4	5	2
C	0	8	1	7
D	2	1	0	1

I have 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 we 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
A	8	4	6	0
B	0	3	5	2
C	0	7	1	7
D	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 that 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 we think about the table, we will see that this is not possible. If we try to come up with a zero cost assignment, we 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 (.4 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
A	8	4	6	0
B	0	3	5	2
C	0	7	1	7
D	2	0	0	1

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
A	9	4	6	0
B	0	2	4	1
C	0	6	0	6
D	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 (.4 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+ l + 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.
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 we 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).

We have done when we reach a stage in which we 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 contain 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 our assignment. Verifying this requires a small argument. I will spare you.

Here is another example.

	1	**2**	**3**	**4**	**5**
A	**81**	**14**	**36**	**40**	**31**
B	**20**	**31**	**25**	**26**	**81**
C	**30**	**87**	**19**	**70**	**65**
D	**23**	**56**	**60**	**18**	**45**
E	**12**	**15**	**18**	**21**	**10**

ISSN- 2394-5125 VOL 7, ISSUE 12, 2020

I will first subtract the minimum element in each row:

	1	2	3	4	5
A	**6**	**0**	**22**	**2**	**17**
B	**0**	**11**	**5**	**6**	**61**
C	**1**	**68**	**0**	**5**	**46**
D	**5**	**38**	**42**	**0**	**27**
E	**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
A	67	0	22	26	0
B	0	11	5	6	44
C	11	68	0	51	29
D	5	38	42	0	10
E	0	3	6	9	71

This array does not permit a zero-cost solution (both 2 and 5 must be matched with .4). Hence we need to change it.

	1	2	3	4	5
A	**7**	**0**	**25**	**2**	**0**
B	**0**	**8**	**5**	**6**	**41**
C	**1**	**65**	**0**	**5**	**26**
D	**5**	**35**	**42**	**0**	**7**
E	**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.

REFERENCES :

- Reeb, J. and S. Leavengood. 1998. Using the Simplex Method to Solve Linear Programming Maximization Problems, EM 8720. Corvallis: Oregon State University Extension Service.
- Hillier, F. S. and G. J. Lieberman. 1995. Introduction to Operations Research, 6th ed. New York: McGraw-Hill, Inc.

ISSN- 2394-5125 VOL 7, ISSUE 12, 2020

- Ravindran, A., D. T. Phillips, and J. J. Solberg. 1987. Operations Research: Principles and Practice, 2nd ed. New York: John Wiley & Sons.
- Dykstra, D. P. 1984. Mathematical Programming for Natural Resource Management. New York: McGraw-Hill, Inc.
- Lapin, L. L. 1985. Quantitative Methods for Business Decisions with Cases, 3rd ed. San Diego: Harcourt Brace Jovanovich.
- Bierman, H., C. P. Bonini, and W. H. Hausman. 1977. Quantitative Analysis for Business Decisions. Homewood, IL: Richard D. Irwin, Inc.