Course Name - Operations and Revenue Analytics

Professor Name - Prof. Rajat Agrawal

Department Name - Department of Management Studies

Institute Name - IIT, Roorkee

Week - 07

Lecture - 32

Welcome, friends. In our last session, we discussed revenue optimization in the case of network problems. We discussed that various service organizations offer services in the context of networks. We discussed the concept of hub and spoke, which is becoming very popular in the airline industry, and we also discussed how this network problem can be seen in hotels, car rentals, and various other services that extend beyond one period. So, whenever there is a single-period issue or a single flight, network optimization may not be required.

But whenever you require services that extend beyond one period or a single flight, you encounter the issue of network optimization. In the case of network optimization, we discussed that a very simple way is the greedy algorithm. We discussed how you make a decision to accept or reject a particular booking request based on the probabilities of demands not occurring at that particular time, and what the future probabilities of these demands could be. Then we also discussed the application of linear programming, and in linear programming, we discussed how you can formulate your problem into linear programming by allocating appropriate capacity to all possible combinations. We discussed a very simple example in our previous session, where we had two legs.

So, you could have customers for individual legs and also customers who may expect services for the entire network. And we formulated a simple LP problem. We discussed how that LP can be solved using any type of linear programming software available. There are many packages available in the market. Extending that discussion, we explored how linear programming can actually help.

In this particular session, we are going to discuss further improvements of that. That is the virtual nesting and the concept of indexing for solving our network optimization problems. Now, before we go to virtual nesting and indexing, let me give you the perspective that whenever we are doing linear programming, there are multiple, you can say, legs which are possible. Like in our case, we discussed Mumbai, Delhi, and Roorkee. In our previous class, to explain this concept of how LP is going to work, we considered a single fare for all these classes.

You may remember that we considered 200 for Mumbai-Delhi. We considered 160 for Delhi-Roorkee. And we considered a single fare from Mumbai to Roorkee via Delhi as 300. But in reality, there may be multiple fare classes for each of these sectors. For example, let us say Mumbai-Delhi, Delhi-Roorkee, and Mumbai-Delhi-Roorkee.

You may have full fare, discounted fare, full fare, discounted fare, full fare, discounted fare. These are all the different possibilities, and therefore, like in our previous problem, we took only three decision variables: x, y, and z. The capacity of this flight is 100 seats, and this flight was 90 seats. You may remember that we had x equals to 80, y equals to 70, and z equals to 20. This was the answer we got for that particular problem. But if I have full fare, discounted fare, full fare, discounted fare, and so on, you may have so many new variables: x_1, x_2, y_1, y_2, z_1, z_2. So now you will have six decision variables for solving.

And let me tell you that in airlines, as well as in hotels generally, for each of these sectors or each of the room categories, you may have more than 10 different types of fare classes. And therefore, it is not going to remain such a simple, small linear programming problem; it is going to become a very complex linear programming problem. For example, I have some fares with me—let us say 150, 100, this is 120, 80, this is 250, and 170. And when we solve such types of problems, you will see that out of these 6 decision variables, there will be some demand for each of these categories you will know. You will find three categories of solutions.

Category 1—and I leave it to you to solve and verify the outcome which I am suggesting. Category 1, where you will have full allocation. The meaning of allocation is we are expecting the demand of, let us say, x_1 equals to 30, x_2 equals to 50. There will be some demand expected for y_1 also, y_2 also, z_1 also, z_2 also. So, there will be a few decision variables for which you will have full allocation.

Whatever demand is there, you are going to keep that many seats for that particular decision variable. Category 2: no allocation. Another extreme—you will have zero allocation for some of these decision variables. They may be just one, they may be two, or more than that; the solution will tell, but there may be some decision variables for which there is no allocation. And there will be a third category of allocations, where you have partial allocation.

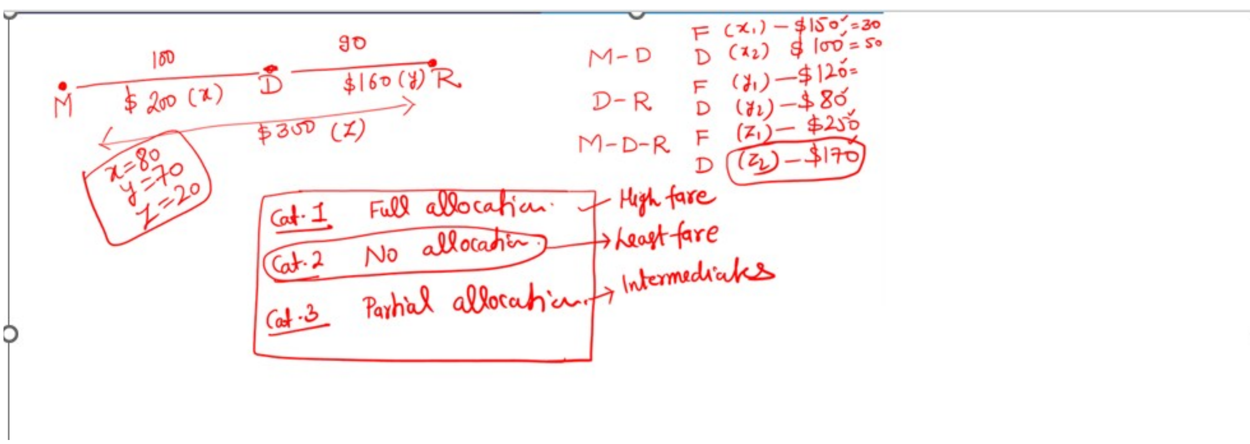
There will be some decision variables out of these six, where you will have allocation but, not as per the expected demand for that particular sector and for that particular fare category. So, all your outcome can be summarized in these three categories and I am leaving it to you to solve this particular in our tutorials, we will give you more data about this so that you can actually solve this particular question. Now, as we know that this problem is going to remain a very complex one. So, for that purpose in this particular session, we are going to focus more on this concept of virtual nesting. Now, this virtual nesting is a very interesting way which is in a way extension of our EMSR concept which we have already studied, when we have only one single flight or one single night of room stay or you are going to rental your car for just for one day.

So, in this virtual nesting concept, we are going to use this process of indexing, indexing we are going to discuss in this particular session only. To group ODFs, ODF is origin destination fares. So, we are going to put some of the origin destination fares of similar values into a bucket. So, there are three, four interesting term which is coming in this particular sentence indexing, ODF and bucketing. Indexing is basically bucketing of ODFs.

So, rather handling all the fare classes differently, we are going to handle them into some buckets. So, we will be making some buckets for this particular example itself and that will help you to understand the practical implementation of this particular solution approach. So, as you remember in a single resource case we said that 1 is the highest fare class, 2 is the second lower and nth class is the cheapest fare class. Similarly, the same convention we are going to follow here and you may remember that highest fare class has the excess of all the capacity that is the concept of nesting. So, same is applicable here also that bucket 1 has the excess of entire legs capacity.

Bucket 2 will have access to the remaining inventory after bucket 1 and so on. So, buckets are similar to the class we had in our single resource cases. Each buckets booking limit and the protection level are updated with each booking or cancellation using EMSR or similar approaches. Like, whenever we are going to have a booking accepted, we are going to reduce the available capacity as per the buckets case, whether it is bucket 1, bucket 2, bucket 3. So, we are going to reduce the capacity from all from the entire leg.

And whenever there is a cancellation, you are going to add the capacity in the entire leg. So, the same thing which we have done in the EMSR. So, therefore, this nesting is similar to the nesting which we have discussed only for single product or single class case. So, to start the process of virtual nesting, first we are going to map every ODF to a bucket on each leg. We are going to estimate the mean and standard deviation of forecasted demand per leg bucket.



And then we are going to use the concept of EMSR for setting the booking limits and protection label for each of these buckets. So, for each booking request, we have to check the buckets capacity as we used to do for a particular class in the earlier cases. We are going to accept the booking request if sufficient capacity is available otherwise, we are going to reject that and if we are accepting a particular booking we have to reduce the buckets available capacity on all legs in the booked ODF. So, and this process will keep on going because periodically re-forecasting ODF's demand and update expected bucket demand per leg that is going to be available to us on a periodic basis. So, whenever the new ODF demand will be available to us, so the mean demand and standard deviation will change and accordingly the available booking limits and protection level for each bucket will also need to be updated.

And because of this new forecast, you have to recalculate your nested booking limits for each bucket based on the new forecast and remaining capacities. So, let us see with the help of this example how this virtual nesting is going to work. So, this is the same example which we are continuing from our previous session about the Mumbai-Delhi-Roorkee sector. Now, in this Mumbai-Delhi-Roorkee sector, we are offering two fare classes, Y class and M class, for each product. So, you see that there are three products.

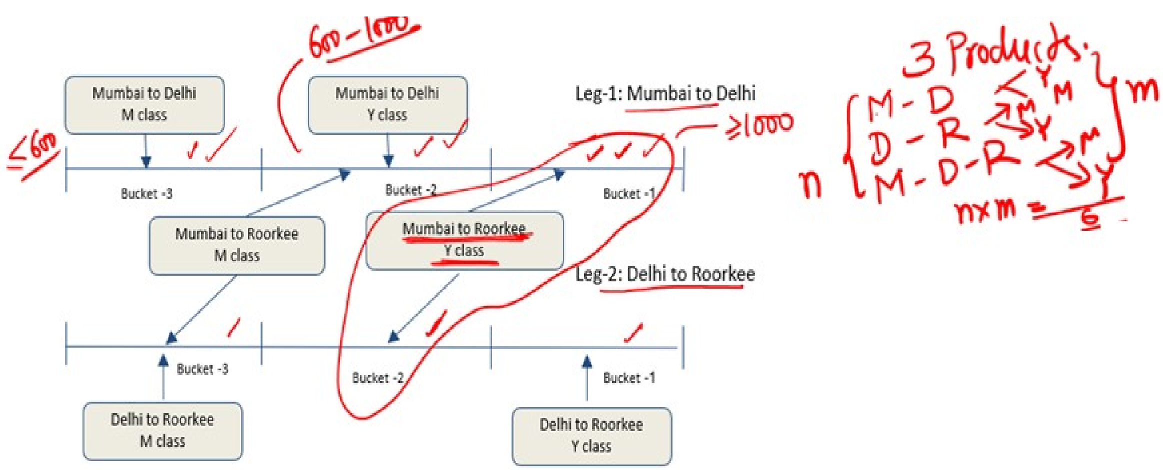
Three products—before I write these products, you should think about what these three products are. The three products are Mumbai-Delhi (that is product number one), Delhi-Roorkee (product number two), and Mumbai-Delhi-Roorkee (that is product number three). And we have two fare classes in each of these products. Y and M, Y and M, Y and M, and YM—these are the two fare classes. So, a total of six options are available.

So, these are N, these are M. So, N into M—these are the total six options available for me. Now, I have to put them into some number of buckets. So, you see how we can do this. These are the first leg, that is Mumbai to Delhi. This is the second leg, that is Delhi to Roorkee.

Now, in Mumbai to Delhi and Delhi to Roorkee, we are dividing each of these into three buckets. Bucket 1, bucket 2, bucket 3. Similarly, bucket 1, bucket 2, bucket 3. Now, in bucket 1, what are the, because you have 6 types of passengers. Passengers travelling only between Mumbai to Delhi in Y class fare, Mumbai to Delhi only in M class fare.

So, in bucket 1, customers who are travelling from Mumbai to Roorkee, Mumbai to Roorkee, in the Y class. These are available in my bucket 1 of Mumbai to Delhi leg and bucket 2 in Delhi to Roorkee leg. As we just explained you that bucket 1 means the entire capacity available in this leg is available to these customers. So, entire capacity available in Mumbai to Delhi leg will be available to these Mumbai to Roorkee Y class customers. While for Delhi to Roorkee leg, it is not the complete capacity available to them, the total capacity minus bucket 1 will be available to these customers.

So, this is how, now rather talking of individual legs, we are going to talk in terms of this kind of bucketing system that what different types of buckets are available. Similarly, for bucket 2, you have customers who are travelling from Mumbai to Delhi in Y class and Mumbai to Roorkee in M class also. So, they all will have access to bucket 2 of the capacity of leg 1, while they will have access to bucket 3 of leg 2. Bucket 1 of leg 2 will be available to Delhi to Roorkee Y class customers only. So, this particular division of capacities into the buckets and according to that you can decide that if a customer is coming in a particular category of fare for a particular sector, you will take a decision of



accepting or rejecting his request.

And similarly, if whenever a booking is accepted, the relevant capacity that whether you are accepting for 1 or whether you are accepting for 5, the relevant seats need to be reduced from the entire leg from wherever you are doing the booking. And whenever you are accepting any cancellation request, you have to update the seats. So, that is virtual nesting. Now, important thing will be that how have we clubbed these particular buckets for particular fare classes. So, that is done with the similar ODFs that you are getting similar ODF for this customer here and this customer here.

Like, you see bucket 2 in leg 1 and bucket 3 in leg 2. These two buckets you are giving to two different types of customers. Bucket 2 in leg 1 is available to Mumbai to Delhi Y class customers as well as Mumbai to Roorkee M class customers that is in bucket 2 leg 1. Bucket 3 in leg 2 is available to Mumbai to Roorkee M class customers and Delhi to Roorkee M class customers also. So, one bucket does not mean that it is only limited to a particular group of customers, one bucket may have access to more than one type of customers.

Because, on the other side if you see bucket 3 for leg 1, it is accessible to only one type of customer that is Mumbai to Delhi M class customers. So, one bucket may be applicable, may be accessible to more than one type of customers or it may be accessible to only just one type of customers. Now, using this virtual nesting, we also need to understand the concept of indexing. Because this bucketing and assigning ODFs to a particular bucket is known as indexing and for understanding the concept of indexing we also need to understand the concept of opportunity cost that is important in this particular context. So, this is very important because if you are assigning a particular ODF value, how are you calculating that ODF value?

In the beginning of this particular session, we had that you have 150, 100, 120, 80, 250 and 170. Now, when you have so many different types of fare classes, if I am giving a particular seat for let us say customer who is travelling in this particular category 170 directly from Mumbai-Delhi-Roorkee. I am giving this particular seat at the expense of some other customer. So, what is the opportunity cost which I am incurring for losing a seat for that customer in favour of this particular customer. So, in all these decisions there are possibilities that there may be some other customer looking for the similar kind of a seat coming from a different set of requirements.

So, we have to understand that concept and that is the opportunity cost of all those customers to whom we are not able to serve because of fulfilling this particular requirement. If I am fulfilling a requirement of a direct segment obviously, this direct segment requirement is fulfilled at the cost of two customers who are looking for individual tickets, who are looking for individual tickets? So, I could have earned some profit from them. But, now I have deprived myself from earning that profit because, I have sacrificed two customers for one single customer and this is the concept of opportunity cost. So, we are now coining a new term which is known as net leg fare.

Rather the revenue or the price you are getting from a particular booking in place of that it is better to have this concept known as net leg fare.

$$\begin{gathered} Net\ leg\ fare\ on\ ODF\ i\ for\ leg\ k = total\ fare\ for\ ODF\ i \\ -Sum\ of\ opportunity\ costs\ on\ \mathbf{all\ resources\ other\ than\ k\ (if\ any)\ in\ ODF\ i} \end{gathered}$$

That what is the final net economic gain to you for giving a booking to a particular customer because, you have sacrificed booking of some other customer. In this example, if I am taking a thorough booking from origin to destination I am sacrificing two customers. But for example, if I take one extra booking in my discounted class, it is quite possible this extra booking in a discounted class is at the cost of high fare or full fare customer. So, this is going to be a loss for me and whenever I am taking one extra customer for the sake of lower fare customer, it is always a profitable because you are going the opportunity cost is much lower and therefore, it is beneficial to always book if customer for higher fare class as compared to lower fare class.

And therefore, if you go back to the discussion we had at the beginning of this session, there may be some categories where you will have no allocation. Because there is no advantage to you in allocating some seats for low-fare or low-revenue customers if you lose some of the seats for higher-fare customers. So generally, full allocation you will see in high-fare or full-fare cases. No allocation, generally, you will see in the lowest-fare cases. Out of all these categories, six categories if I consider in order of their increasing or decreasing.

So, you will not see any allocation where your revenue expectation is minimal, and for intermediates, you will have some kind of partial allocations. So, in this way, this allocation problem is seen. So, this net leg fare from one origin to another means, for a particular category ODF (origin-destination fare) for a particular leg, total fare for ODFI minus the sum of opportunity costs in all resources other than K, if any, in that particular ODI. So, with the help of this, you can again do the classification of your bucketing problem. Again, taking the example similar to that here, we have the opportunity cost given to you—the complete opportunity cost—that is, for Mumbai to Delhi for leg 1, the opportunity cost is 250, and for leg 2, the opportunity cost is 800 dollars.

Now, in each leg, as already discussed, we have three buckets: bucket 1, bucket 2, and bucket 3. Now, these buckets are classified on the basis of similar ODF values, and these values are: bucket 1, where our net leg fare is more than 1000; bucket 2, 600 to 1000; and bucket 3, where net leg fare is less than 600 dollars. So, if I go back to this particular example, bucket 1, where our net leg fare is 1000 and more; this is 600 to 1000; and this is less than 600. So, in this way, considering some economic values of the buckets, you have classified them on the basis of their ODF values, and therefore, you will have a particular type of benefit when you are allocating to a particular bucket class. And in this way, our virtual nesting will help us apply the concept of EMSR for our booking of network optimization problems.

In our next class, we will try to do some numerical exercises so that we can show you the live calculation of these net leg fares. And, on the basis of bucketing, we will see how we have allocated the limited capacity into different types of buckets and how we are going to make decisions based on that. So, with this, we come to the end of this particular session where we understood the theoretical aspects of virtual nesting and indexing for our network optimization problem. Thank you very much.