So far, we've only considered optimizing the fares for a single route.

In this video, we'll change our optimization formulation to include connecting flights.

Now, instead of just being able to go from JFK in New York to LAX in Los Angeles, let's suppose that the plane stops in Dallas at the Dallas Fort Worth airport.

We still are just using one plane, but the passengers can now fly from New York to Dallas, Dallas to Los Angeles, or from New York to Los Angeles by just staying on the plane in Dallas.

So how does our optimization problem change?

We now have six types of seats that we can offer: the original two types, regular and discount from New York to LA, and four new types.

We can sell both regular and discount seats from New York to Dallas, and regular and discount seats from Dallas to Los Angeles.

We know the price of each type of ticket as well as the forecasted demand for each type of ticket.

We also know that we have a capacity of 166 seats on our plane for each leg of the trip.

There's room for 166 passengers on the plane from New York to Dallas, or the first leg of the trip.

Then the passengers with a final destination of Dallas will get off the plane and the passengers flying from Dallas to LA will get on the plane.

On the second leg of the trip, flying from Dallas to LA, we also have a capacity of 166 seats.

So we need to remember that the passengers flying from New York to LA will take up capacity on both legs of the trip, while the other types of passengers will only take up capacity on one leg of the trip.

So what are our decisions now?

They're the number of regular tickets to sell for each type, and the number of discount tickets to sell for each type.

So in total, we have six decisions to make.

Now, let's define our objective.

Like before, it's to maximize the total revenue.

This is the sum of the price of the ticket times the number of seats of that type we sell, for each type of ticket.

And like before, we have two types of constraints-- capacity constraints and demand constraints.

For the capacity constraints, the airline shouldn't sell more seats than the capacity of the plane, for each leg of the trip.

So we need two capacity constraints here: one for the New York to Dallas leg and one for the Dallas to LA leg.

Note that the New York to LA passengers have to be counted on both legs of the trip.

So the first constraint accounts for all passengers that need to be on the plane when it flies from New York to Dallas, and the second constraint accounts for all passengers that need to be on the plane when it flies from Dallas to LA.

We also need six demand constraints, one for each type of ticket.

The number of seats sold should not exceed the forecasted demand for each type.

And lastly, we can't sell a negative number of seats, so we have our non-negativity constraints to prevent the variables from being negative.

Let's now go to LibreOffice and adjust our formulation to solve this bigger problem.

In LibreOffice, go ahead and open the file Week9_AirlineRM_Connecting.ods.

In this file, I've set up our data, our decisions, our objective, and our constraints.

Our decisions, again, are highlighted in yellow.

We have a decision for each type of seat on each flight.

Our objective here is the spot in blue.

To build our objective, we'll use the sumproduct function.

So type = and then sumproduct, and in parentheses, select all six prices, type a semicolon, and then select all six decisions.

Close the parentheses and hit Enter.

We see here, like we did before, that we have 0 in our objective, because right now, we're not selling any seats.

Now let's create our constraints.

The first constraints are capacity constraints.

The first is the capacity on the leg from New York to Dallas.

The left-hand side should be equal to the seats from New York to LA plus the seats from New York to Dallas.

The sign is less than or equals and the right-hand side is 166, the capacity of our aircraft.

Now we need to build the capacity constraint from Dallas to LA.

The left-hand side is equal to the seats from New York to LA plus the seats from Dallas to LA.

Our sign is, again, less than or equals and our right-hand side is 166.

For the demand constraints and the non-negativity constraints, because we have six of each this time, we'll actually make them in a more efficient way than before.

So we just have a note down there that we need to remember to add these constraints.

So now go ahead and in the Tools menu, select Solver.

We need to first fill in the target cell, which should be the objective.

Make sure that Maximum is selected.

Then, in the Changing Cells box, select all six decisions.

Down in the Limiting Conditions, let's now build our constraints.

For the Cell Reference column, let's start by selecting the left-hand side of the two capacity constraints.

The Operator should be less than or equals and the Value should be the right-hand side of these two capacity constraints.

Now let's make the demand constraints.

In Cell Reference, just directly select the six decision variables, make sure the Operator's less than or equals, and for the Value, select the six demand constraints.

This is a bit easier than what we did before because we didn't have to type them all out in our spreadsheet.

Now let's do a similar thing for the non-negativity constraints, where in Cell Reference, we select the six decisions.

The Operator this time should be greater than or equals, and for the Value, just type 0.

Make sure in Options that the Linear Solver is selected, and go ahead and hit Solve.

The solving result should say: "Solving successfully finished.

Result: 120,514." This is our total revenue.

Go ahead and click Keep Result, and let's take a look at our solution.

So we see here that the optimal solution is to sell 80 tickets for the regular price from New York to LA, 0 of the discount price from New York to LA, 75 of the regular price from New York to Dallas, 11 of the discount price from New York to Dallas, 60 of the regular price from Dallas to LA, and lastly, 26 of the discount price from Dallas to LA.

We saw here that we could pretty easily solve a more complicated problem in LibreOffice that we probably couldn't have solved as easily by inspection.