

## MITOCW | MIT15\_071S17\_Session\_9.4.04\_300k

---

Here we are in LibreOffice, and you'll see that we have all the data we discussed in the previous videos loaded into the spreadsheet here.

Let's take a minute to familiarize ourselves with how the data is set up.

So here we see we've got the number of surgical teams per department per day, and we've got it listed by department-- all five of them are here-- and by day of the week-- Monday, Tuesday, Wednesday, Thursday, and Friday.

Remember, for all departments, we have the same number of teams in per day, except for oral surgery, because the oral surgeon is only present on Tuesdays and Thursdays.

Scrolling across, we see that for all five departments, again, we've got the minimum number of operating rooms that each one requires each day.

This is simply 0.

As we keep going, we see the maximum number of operating rooms that each department requires each day.

Going down to the next set of data, we've got the weekly department requirements.

We've got the minimum and maximum number of operating rooms that each department requires each week.

We also have the weekly targets.

This is data, because it is set by the departments before the operating room manager has a chance to schedule the operating rooms.

So this is our problem data, and now let's go down to our decision variables.

All right, so here are our decision variables.

You see that we've set up a decision variable for each department, and for each day of the week.

Right now, they're simply set to 0, but we've colored them yellow, so they'll be easy to see.

Let's now set up a column for weekly totals.

Let's put that here.

All right.

So let's fill this in.

The weekly total for ophthalmology should just be the sum of all of the number of operating rooms assigned to ophthalmology from Monday through Friday, so that's this entire row.

Just like that.

And we can drag this formula down for every department.

OK.

So now, if you actually go through each one of these, you can see up in the Formula tab that we've got the formula for the weekly totals here.

So let's get started by putting in our objectives and our constraints.

So down here, I've set up a box for the objective, but I haven't yet put in a formula for it.

So let's remind ourselves what the objective is here.

The objective is to maximize the percent of target allocation hours that each department is actually allocated.

So how can we calculate that?

Our decision variables up here represent the number of operating room hours that each department is assigned on each day.

We calculated the total over here, as the number of operating rooms that a department is assigned over the course of the week.

To turn that into hours, we need to multiply it by 8, because operating rooms are staffed for eight hours.

So let's create a column that turns this into hours.

We'll put it over here, and we'll call it weekly hours.

OK?

And this will just be equal to 8 times the weekly totals.

Great.

And we'll do this for all departments.

Here, this time I'll actually write it out.

OK.

Great.

So now we've got the weekly hours here, but to calculate what fraction of the weekly target, we've got to divide it by the total weekly target.

And the total weekly target, we see, is up here.

So let's set this up for every department in a column called "Percent of Target".

OK, so this will just be equal to the weekly number of hours, divided by the weekly target.

OK, so we'll just set that up for each department.

Certainly there are faster ways of doing this, but I don't want to speed ahead if you guys aren't yet LibreOffice wizards.

OK.

All right, so we've got the percent of target reached for every department.

So our objective-- we can go back to it now-- it'll just be the sum of all of these percent of targets.

So let's add that in-- equals sum, and then we'll add in all of these percent of targets.

Great.

So right now, of course, we haven't solved the problem yet, so we've just got 0 in our objective.

But before we go ahead and solve it, we need to add in our constraints.

So down here in green, we have a list of all the constraints we outlined in the previous videos.

Let's review them quickly.

So the first one is just that the number of operating rooms assigned each day is integer.

We can't assign ophthalmology  $\frac{3}{4}$  of an operating room on Thursday, or general surgery 6.5 operating rooms on

Monday.

We need to make sure that these take integer values.

The next constraint is just that the total number of operating rooms each day has to be less than or equal to 10.

So while we're here, let's just add in a line underneath our decision variables, which calculates the total number of operating rooms each day.

Over here we'll call it total number of operating rooms each day, and this will just take the sum over all the operating rooms we assign on Monday.

Just like that.

Great.

And let's drag this formula across to use it for the entire week.

OK, so now we've got that set up, so we'll be able to do the second constraint.

The third set of constraints is the upper bound-- that is, the upper bound on the number of surgical teams that are available each day.

We also have an upper bound on the daily department operating room requirement, and a lower bound on the daily department operating room requirement.

Similarly, we have an upper bound and a lower bound on the weekly department operating room requirements.

Our last set of constraints has to do with departmental targets.

We want to make sure we don't give any department more operating room hours than they actually want.

So that means we want to make sure that the number of hours given to each department is less than or equal to the targeted number that they asked for.

All right, so let's go ahead and add these constraints into the Solver.

So we'll start by putting our cursor on the objective function, and going up to the Tools menu.

And here we select Solver.

OK.

So because we started our cursor on the objective function cell, it comes up properly with the "Target cell" already labeled.

We do want a maximum, so we keep that the way it is.

And then here, where it says "By changing cells", we put in our decision variables.

So we just select them all, like this.

All right.

Now, before we start putting in our constraints, I want to remind you to go into Options, and you'll need to change this from DEPS Evolutionary Algorithm to LibreOffice Linear Solver.

OK.

So you notice that the settings-- you can choose to assume variables are integer, and assume variables are non-negative.

We'll do that here.

In fact, this takes care of our first constraint, which was to assume that the variables are integer.

OK.

So we've got this set up.

Now let's put in the rest of our constraints.

OK.

So we've already taken care of this first constraint, that the daily number of operating rooms assigned is integer, and we've taken care of that with our options.

So we'll go down to the next one-- the total number of operating rooms each day.

Remember that we can't assign more than 10, so we'll select this whole row, which has the total number of operating rooms assigned each day, and we'll say that each element in this row has to be less than or equal to 10.

Great.

So on to the upper bound of surgical teams per day.

Let's select our decision variables, all of them at once.

OK.

And then we'll need to scroll up to our data.

OK.

You'll notice here how I'm inputting all these constraints all at once.

We're actually putting in 25 different constraints, one for each day of the week, and one for each surgical team.

And we're just putting them in with a single line, by saying that all our decision variables need to be less than or equal to this box of constraints.

Just like that.

OK.

So our next set of constraints is on the upper bound and lower bound of the daily department OR requirements.

So we'll put in our decision variables again.

Scroll down to those.

And let's say that these are less than or equal to-- sometimes we've got to move the solver around-- the maximum number of ORs per day per department.

OK.

And we'll do that again for the lower-bound.

This time we need to change the operator to greater than.

Greater than the minimum number of operating rooms per department per day.

You see here that we've got 0's in for the minimum number of operating rooms per day, but you could conceivably expect that one department might actually want a minimum number of operating rooms per day.

For example, general surgery, which we see here has a weekly target of 189 hours, might actually want to have

more than 0 operating rooms per day.

They might want to guarantee they've got at least one.

We'll experiment with changing some of these numbers after we solve the initial problem.

For now, let's get our remaining constraints in.

The next ones we have are the weekly department OR requirements-- the upper-bound and the lower-bound.

So let's just scroll down here in the solver to give ourselves a few more rows.

OK.

So here we'll put in our weekly totals.

Good thing we've already got this set up.

OK, so weekly totals-- this will have to be less than the maximum required by any department in a given week.

So in the Value column, we go over here, and we put in the maximum required each week.

We'll do the same for the lower-bound, again, changing the operator to a greater-than or equal-to sign.

OK.

And we're on to our very last set of constraints, which is the departmental targets down here.

So remember, we can't assign a department more hours than they actually wanted to have, because operating-room time is so expensive.

So we'll select there the weekly number of hours, which we already have a column set up to calculate.

So we'll put this in here that the weekly number of hours must be less than or equal to the target allocation hours.

So scroll back up here to the target allocation hours.

OK, great.

Now we've got all our constraints in, and we're ready to go.

So let's hit Solve.

It may take a few seconds-- up to 100.

In the next video, we'll look at the results that we get, and interpret them.

We'll also see how they change, if we experiment with changing some of the numbers in the problem.