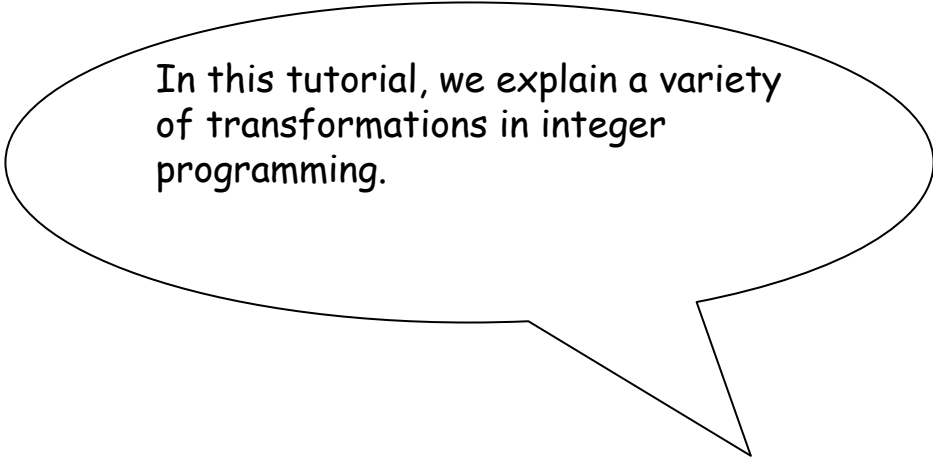


# Transformations in Integer Programming



In this tutorial, we explain a variety of transformations in integer programming.

# But first, a hard question.

Why haven't there been any tutorials in such a long time?

I missed them too. But we have created lots of other educational materials.

**Tim, the turkey**



# Transforming Logical Conditions

$$\text{Max } 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6$$

$$5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14$$

$$x_j \in \{0,1\} \text{ for each } j = 1 \text{ to } 6$$

Here is the integer program used in the first integer programming lecture. As you recall, it's based on a game show that I was on.

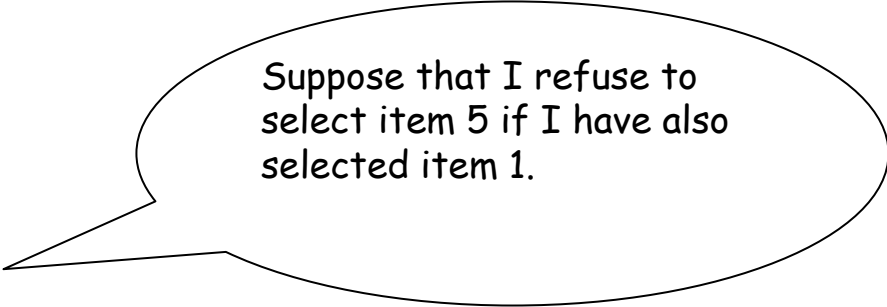
So that we can learn modeling of logical constraints, I'm going to pretend to add conditions on what I will choose.

**Nooz, the most trusted name in fox.**

$$\text{Max } 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6$$

$$5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14$$

$$x_j \in \{0,1\} \text{ for each } j = 1 \text{ to } 6$$



Suppose that I refuse to select item 5 if I have also selected item 1.

In this case, the feasible solutions for Nooz will permit  $x_1 = 1$  or  $x_5 = 1$ , but not both.

This can be modeled via the linear constraint,

$$x_1 + x_5 \leq 1$$

**Nooz, the most  
trusted name in fox.**

I don't get it.  
The linear  
constraint  
doesn't look  
anything like the  
logical constraint.

But how will we  
know that?

Well, Tim. The important thing  
isn't whether it "makes sense" as a  
logical constraint. The important  
thing is that an optimal solution  
for the integer program will  
produce an optimal solution for  
the original problem.

In this case, we need only focus on  
variables  $x_1$  and  $x_5$  to see that the  
constraint works. We don't need  
to think about the other variables.

You see, Nooz just wants to  
eliminate the possibility that  
items 1 and 5 are both selected.  
In other words, we cannot have  $x_1$   
 $= 1$  and  $x_5 = 1$ . The linear  
constraint accomplishes the same  
thing, taking into account that  $x_1$   
and  $x_5$  are both binary.

Tim

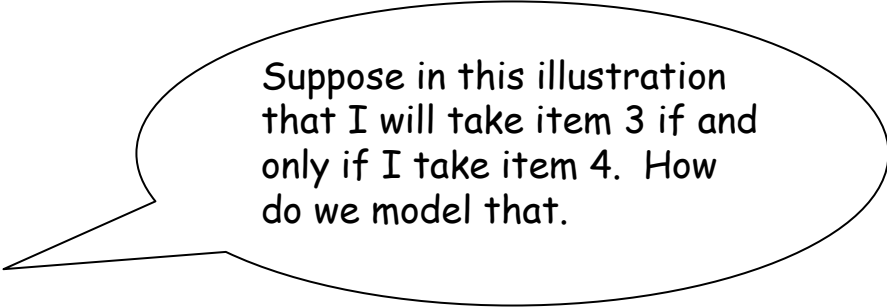
Ollie

# Another logical constraint

$$\text{Max } 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6$$

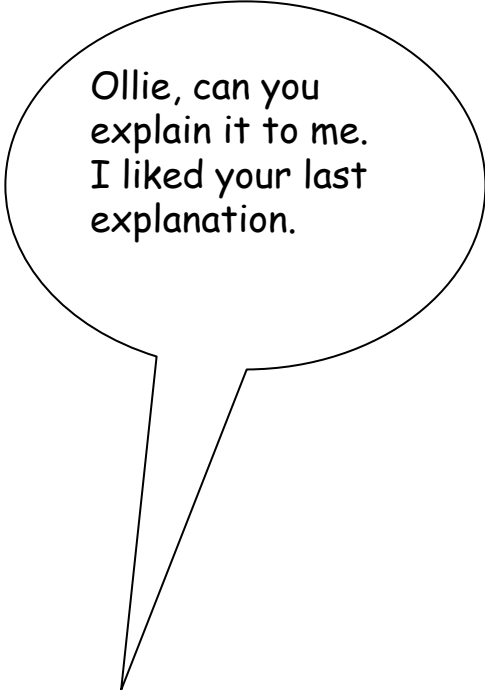
$$5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14$$

$$x_j \in \{0, 1\} \text{ for each } j = 1 \text{ to } 6$$

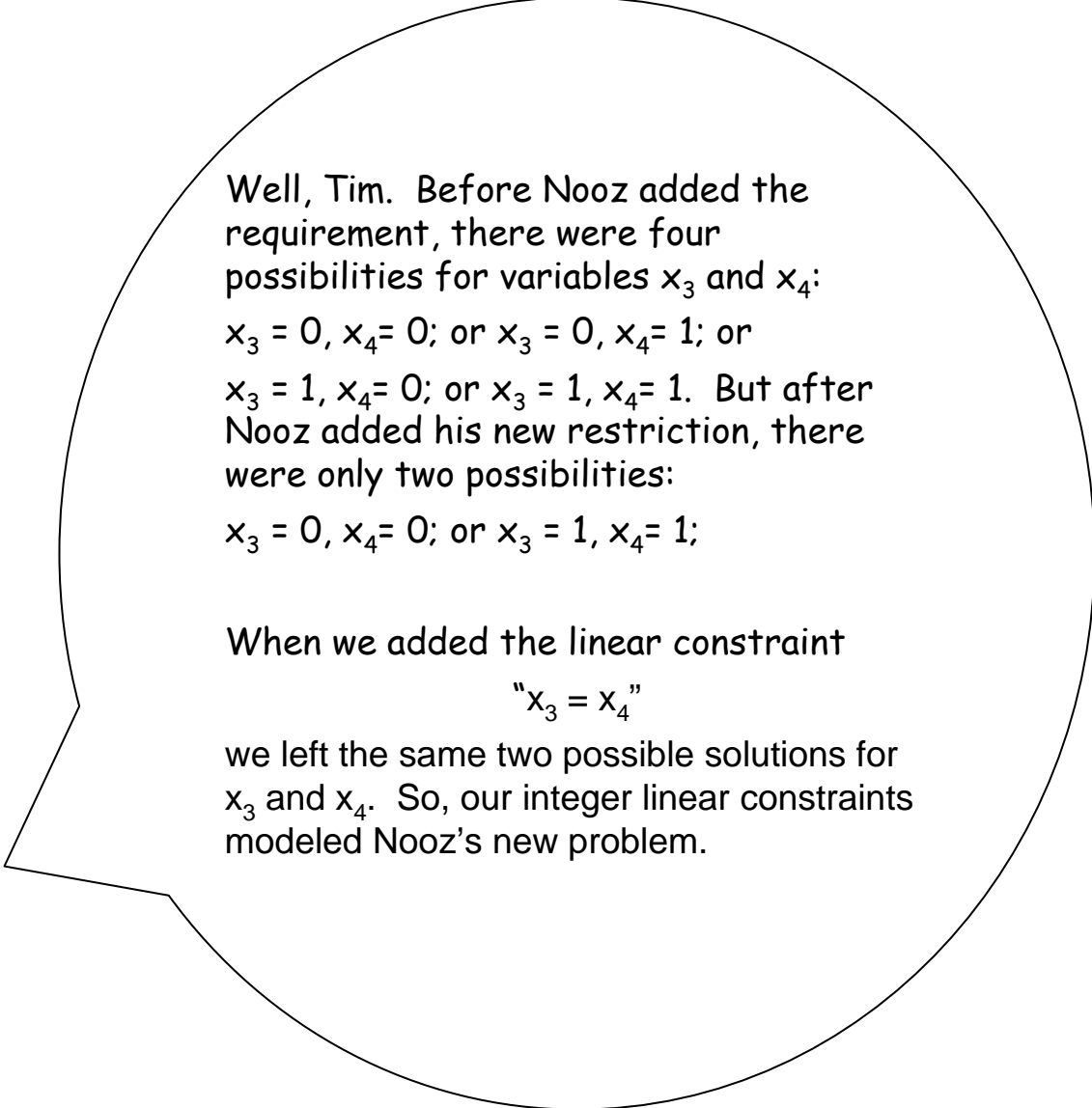


Suppose in this illustration that I will take item 3 if and only if I take item 4. How do we model that.

In this case, the feasible solutions for Nooz will permit  $x_3 = 1$  and  $x_4 = 1$ , or else  $x_3 = 0$  and  $x_4 = 0$ . We can model it as the linear constraint  $x_3 = x_4$ .



Ollie, can you explain it to me. I liked your last explanation.



Well, Tim. Before Nooz added the requirement, there were four possibilities for variables  $x_3$  and  $x_4$ :  $x_3 = 0, x_4 = 0$ ; or  $x_3 = 0, x_4 = 1$ ; or  $x_3 = 1, x_4 = 0$ ; or  $x_3 = 1, x_4 = 1$ . But after Nooz added his new restriction, there were only two possibilities:

$x_3 = 0, x_4 = 0$ ; or  $x_3 = 1, x_4 = 1$ ;

When we added the linear constraint

$$"x_3 = x_4"$$

we left the same two possible solutions for  $x_3$  and  $x_4$ . So, our integer linear constraints modeled Nooz's new problem.

Ollie

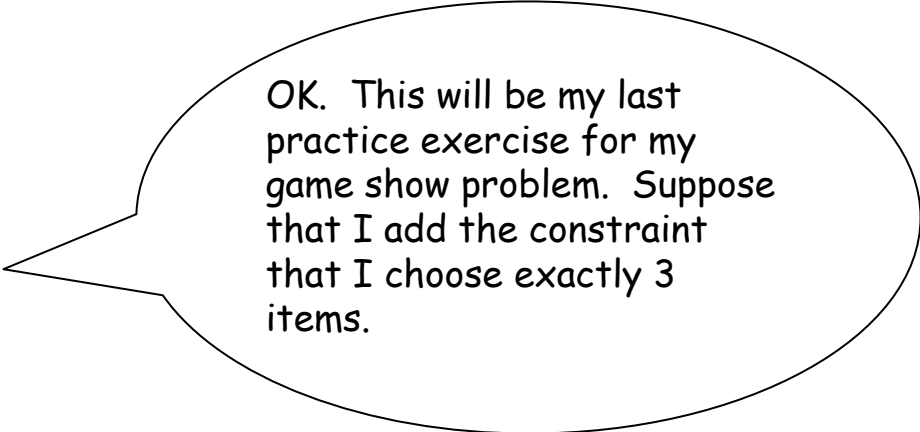
Tim

# One more logical constraint

$$\text{Max } 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6$$

$$5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14$$

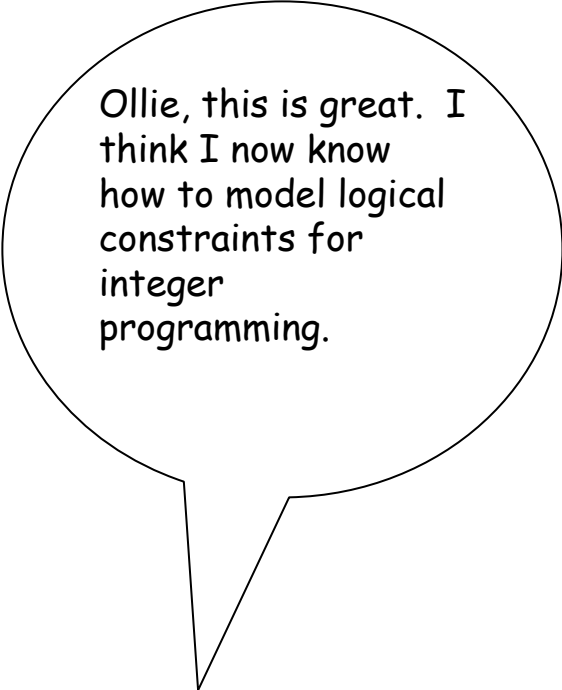
$$x_j \in \{0,1\} \text{ for each } j = 1 \text{ to } 6$$



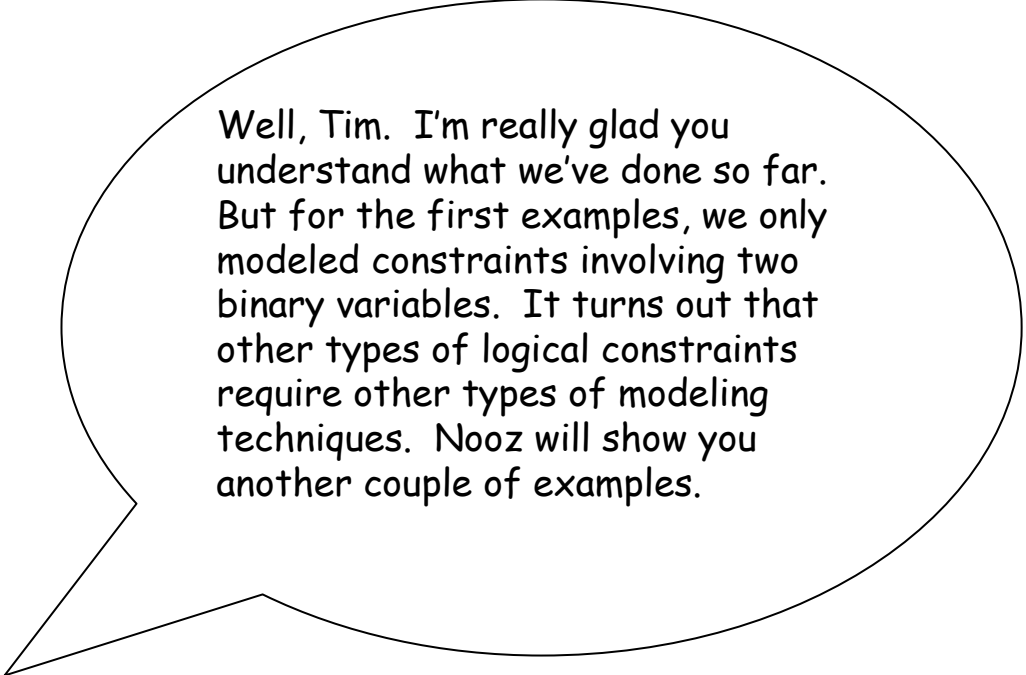
OK. This will be my last practice exercise for my game show problem. Suppose that I add the constraint that I choose exactly 3 items.

In this case, Nooz's constraint sounds like a linear constraint. And it can be modeled with the constraint.

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 3.$$



Ollie, this is great. I think I now know how to model logical constraints for integer programming.



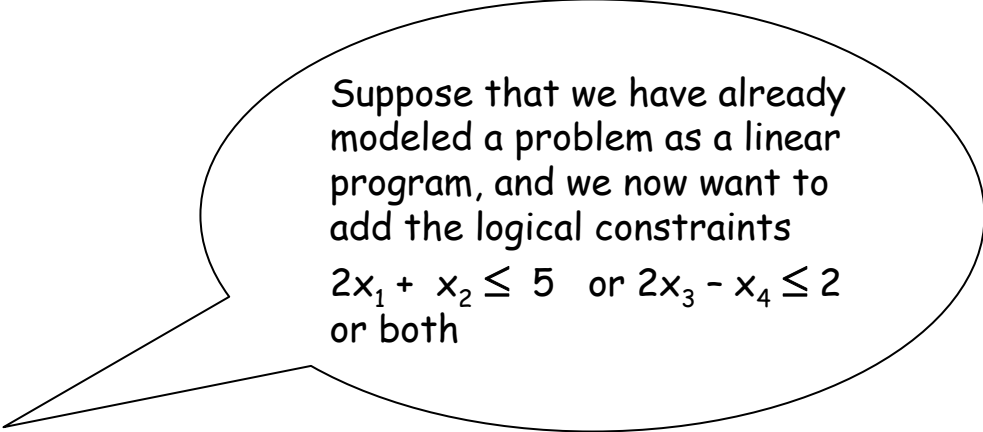
Well, Tim. I'm really glad you understand what we've done so far. But for the first examples, we only modeled constraints involving two binary variables. It turns out that other types of logical constraints require other types of modeling techniques. Nooz will show you another couple of examples.

**Ollie**

**Tim**

# Transforming “Non-Exclusive OR” Constraints

Either  $2x_1 + x_2 \leq 5$  or  $2x_3 - x_4 \leq 2$  or both



Suppose that we have already modeled a problem as a linear program, and we now want to add the logical constraints  
 $2x_1 + x_2 \leq 5$  or  $2x_3 - x_4 \leq 2$   
or both

This situation is much more complicated, but there is a standard technique for doing it.

We are not assuming here that  $x_i$  is binary. In fact, we are not even assuming that it is required to be integer valued. But for our transformation to work, we do need to require it to be bounded. So we will assume that  $x_i \leq 100$  for each  $i$ .

# The Transformation

**Either  $2x_1 + x_2 \leq 5$  or  $2x_3 - x_4 \leq 2$  or both**

In order to model this “either-or” constraint, we add a variable  $y_1$ , which is required to be binary, and we add two new constraints to our original linear program.

$$2x_1 + x_2 \leq 5 + My_1$$

$$2x_3 - x_4 \leq 2 + M(1 - y_1)$$

$$y_1 \in \{0, 1\}$$

where  $M$  is a constant that is sufficiently large. In this case, we could choose  $M$  to be 300 or anything larger.

Either  $2x_1 + x_2 \leq 5$  or  $2x_3 - x_4 \leq 2$  or both

$My_1$

$$2x_1 + x_2 \leq 5 +$$

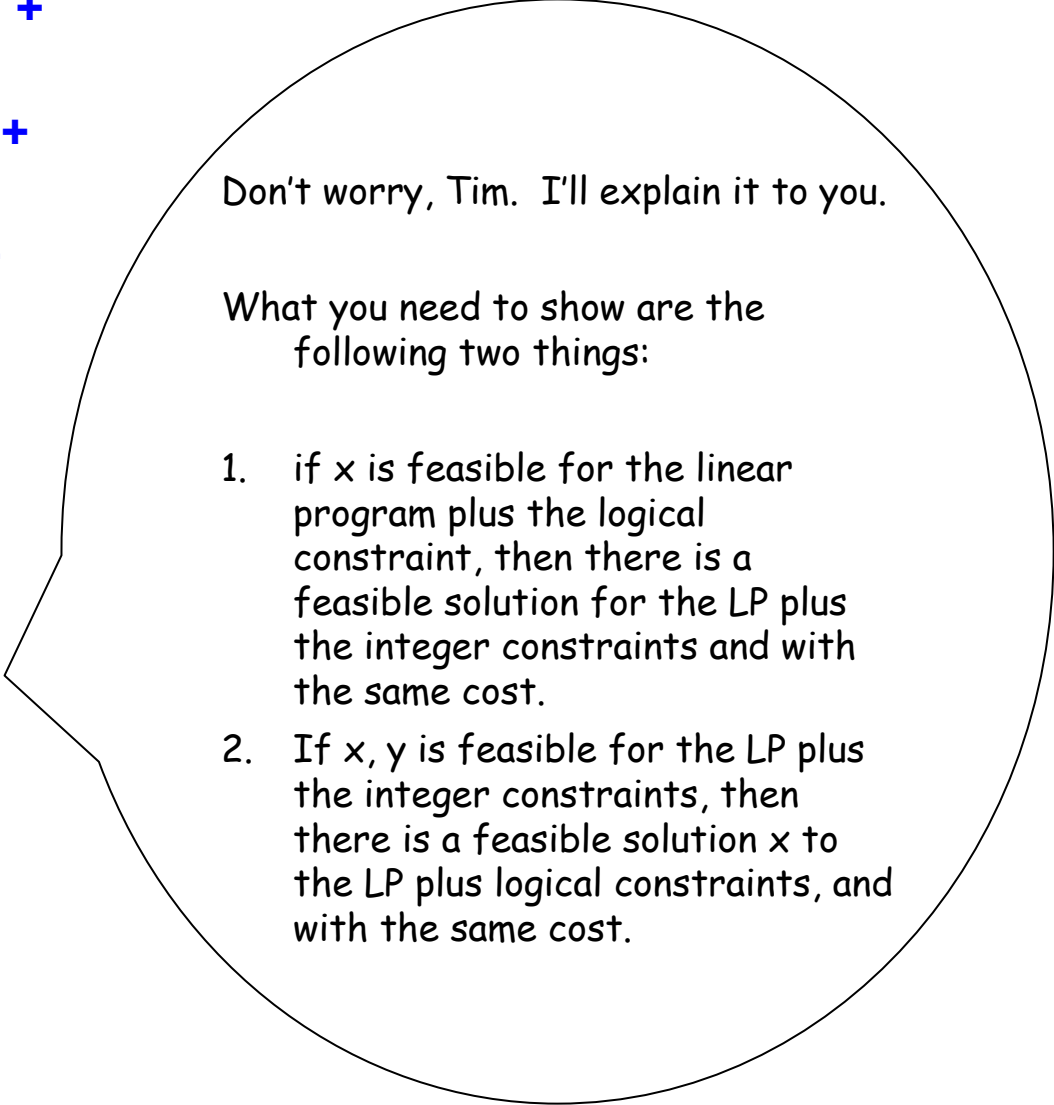
$M(1 - y_1)$

$$2x_3 - x_4 \leq 2 +$$

$$y_1 \in \{0, 1\}$$



I'm lost again.



Don't worry, Tim. I'll explain it to you.

What you need to show are the following two things:

1. if  $x$  is feasible for the linear program plus the logical constraint, then there is a feasible solution for the LP plus the integer constraints and with the same cost.
2. If  $x, y$  is feasible for the LP plus the integer constraints, then there is a feasible solution  $x$  to the LP plus logical constraints, and with the same cost.

Ollie

Tim

Either  $2x_1 + x_2 \leq 5$  or  $2x_3 - x_4 \leq 2$  or both

$$2x_1 + x_2 \leq 5 + My_1$$

$$2x_3 - x_4 \leq 2 + M(1 - y_1)$$

$$y_1 \in \{0, 1\}$$

But that seems like so much to have to remember and so much to show.

Tim, it's the essence of modeling and also of transformations. Whenever you have a real life problem that you model, you want to show that your model matches the real life problem. Usually this takes two parts. First you show that any feasible solution to your real life problem gives a feasible solution to your model, and with the same cost. Then you show that any feasible solution to your model gives a feasible solution to your real life problem and with the same cost.

Often we don't bother showing it for linear programs, because it seems obviously true. But with integer programs, it's not so obvious, and it helps to show it.

Ollie

Tim

**Either  $2x_1 + x_2 \leq 5$  or  $2x_3 - x_4 \leq 2$  or both**

$$2x_1 + x_2 \leq 5 + My_1 \quad (1)$$

$$2x_3 - x_4 \leq 2 + M(1 - y_1) \quad (2)$$

$$y_1 \in \{0, 1\}$$

So, let us first consider the case that  $x$  is feasible for the linear program plus the logical constraint. We first consider the case that  **$2x_1 + x_2 \leq 5$** .

For the integer program, we let  $y_1 = 0$ . Since  $x$  satisfies the logical constraint, it also satisfies (1) which is equivalent to

**$2x_1 + x_2 \leq 5 + 0$** . In addition, since  $x$  is feasible for the linear program, it follows that  $x_3 \leq 100$ , and  $x_4 \leq 100$ , and therefore

$$\mathbf{2x_3 - x_4 \leq 200}.$$

**Therefore  $2x_3 - x_4 \leq 2 + 300(1 - y_1) = 302$**  and (2) is satisfied.

You can now see why we needed  $x$  to be bounded, and why we had a large constant value for  $M$ . We wanted ensure that constraint (2) was satisfied whenever we let  $y_1 = 0$ .

**Either  $2x_1 + x_2 \leq 5$  or  $2x_3 - x_4 \leq 2$  or both**

$$2x_1 + x_2 \leq 5 + My_1 \quad (1)$$

$$2x_3 - x_4 \leq 2 + M(1 - y_1) \quad (2)$$

$$y_1 \in \{0, 1\}$$

So, we next consider the case that  $x$  is feasible for the linear program plus the logical constraint and that

$$2x_3 - x_4 \leq 2$$

For the integer program, we let  $y_1 = 1$ . Since  $x$  satisfies the logical constraint, it also satisfies (2) which is now equivalent to

$2x_3 - x_4 \leq 2 + 0$ . In addition, since  $x$  is feasible for the linear program, it follows that  $x_1 \leq 100$ , and  $x_2 \leq 100$ , and therefore

$$2x_1 + x_2 \leq 300 .$$

**Therefore  $2x_1 + x_2 \leq 5 + 300 y_1 = 305$ , and (1) is satisfied.**

We could have chosen a slightly smaller value of  $M$ . In addition, any larger value of  $M$  would have worked.

Either  $2x_1 + x_2 \leq 5$  or  $2x_3 - x_4 \leq 2$  or both

(1)

$$2x_1 + x_2 \leq 5 + My_1$$

(2)

$$2x_3 - x_4 \leq 2 + M(1 - y_1)$$

$$y_1 \in \{0, 1\}$$

Wow. That was a lot. But now we're done.

No, Tim. We still haven't shown that a feasible solution for the LP plus integer linear constraints is feasible for the LP plus logical constraints. But this part is easier.

If  $x, y$  is feasible for the LP plus integer, linear constraints, then  $y_1 = 0$  or  $y_1 = 1$ . In the first case, the logical constraint is satisfied because (1) is equivalent to  $2x_1 + x_2 \leq 5$ . In the second case, the logical constraint is satisfied because (2) is equivalent to  $2x_3 - x_4 \leq 2$ . Now we are done with the proof that the transformation works.

Ollie

Tim

# Transforming If-Then Constraints

If  $2x_1 + x_2 \leq 5$  then  $2x_3 - x_4 \leq 2$ .

Suppose that we have a linear program, but we want to add the constraint

"If  $2x_1 + x_2 \leq 5$  then  $2x_3 - x_4 \leq 2$ ".

How can we do this using integer plus linear constraints?

This situation is very similar to the previous one, because the "if-then constraint" is equivalent to writing

$2x_1 + x_2 > 5$  or  $2x_3 - x_4 \leq 2$  or both.

But there is an added complication. We don't like a strict inequality constraint. So, we will show how to carry out the transformation in the special case that  $x_1$  and  $x_2$  are integer valued. In this case

$2x_1 + x_2 > 5$  if and only if  $2x_1 + x_2 \geq 6$ .

## Transforming the “if-then” constraint.

If  $2x_1 + x_2 \leq 5$  then  $2x_3 - x_4 \leq 2$   
is equivalent in this case to

$2x_1 + x_2 \geq 6$  or  $2x_3 - x_4 \leq 2$  or both.

This is equivalent to

$$\begin{aligned}2x_1 + x_2 &\geq 6 - My_1 \\2x_3 - x_4 &\leq 2 + M(1 - y_1) \\y_1 &\in \{0, 1\}\end{aligned}$$

But this time, we leave it as an exercise to the student to fill in the details of why it works. As before, we assume that all variables are bounded by 100. In this case, we can let  $M$  be 200 or higher.

By the way, if  $x_1$  and  $x_2$  were not integer valued, we don't know how to model the “if then constraints” correctly using integer and linear constraints.

## Last Slide

There are lots of other techniques used in modeling for integer programming. But that is all for this tutorial.

I hope that you found this tutorial useful. And I hope to see you again soon.

**Cleaver**