Readings for Assignment 3


*See especially the Wyndor setup cost example at the end of that chapter. The key to that problem is described below in this Study Guide.*

Main Topics for Assignment 4

- In this section of the course, we introduce the idea of binary integer variables. That means the changing cells are constrained to be binary. The value of a changing cell can only be 0 or 1. No other values are permitted.

- Binary integer programming (BIP) with zero-one variables makes new techniques and tricks possible, while maintaining a linear model. With binary variables we can model yes-no decisions, or mutually exclusive decisions, or even contingent problems with an if-this-then-that feature. Such decisions sound nonlinear, as they will have branching or step-function characteristics. But binary variables do allow us to model many such things with a linear model.
  - Linear models are easier to solve. You should keep your models linear whenever possible.

- Solver also will accept a general integer constraint on the changing cells. Sometimes the changing cells, realistically, always should end up as integers. A model may be solving for the number of people, for example, or the number of trucks. The solution shouldn't include fractions of things that actually are indivisible.
  - Models with integer constraints behave differently, however, and can be harder for Solver to solve.

Revised Due Date

This assignment will be due **Wednesday evening, June 8**. You must upload your completed file to the Assignment 4 page on the K510 Canvas site by 11:59 pm Wednesday night to avoid late penalties. Please do NOT email your file to me. Just upload it to Canvas.
Practice Problems

Several examples of binary IP models are posted on Canvas:

1. The California Manufacturing problem, at the beginning of Chapter 7, illustrates a mutually exclusive decision. There are two possible locations for the new warehouse: Los Angeles or San Francisco. But the company only needs one warehouse. So whichever city gets chosen, the other city must NOT be chosen. The warehouse can’t be split between the two locations.

2. The electricity generator problem demonstrates the use of binary changing cells in choosing which generators to use, given that each one has a startup cost if used. I have made a short video on this one, walking through the construction of the spreadsheet.

3. The airplane production problem is another example of a fixed startup cost only if a certain option is chosen.

4. The sales rep assignment problem is fairly complicated, but I’ve annotated the spreadsheet with hints and tips for building the model. This problem includes a little more sophisticated way to use binary variables on a capacity constraint to create an “upper bound.”

See if you can design your own spreadsheets for these models, before you look at my solutions. Notes on these topics are included below in this Study Guide.

Notes on Integer Programming Models in General

In the linear models we have dealt with to this point, the decision variables in the changing cells could take on any values (or at least, any non-negative values). So the changing cells could end up being fractions. In some situations, however, such as the number of people, fractional values don’t make sense. In those situations, it is possible to constrain the changing cells to be integers.

The problem still can be solved as a linear model. You should still select the Simplex LP solving method (or, in older versions of Excel, activate the option for “Assume Linear”). In the constraint menu in the Solver dialog box, select the option for “int” from the drop-down box. That selection causes Solver to put only integers in the changing cells.

In addition, don’t forget this one: In the Options page for Solver, be sure to uncheck the option for “Ignore integer constraints.” This is turned on by default, so in any IP or binary model, you must specifically UNCHECK the option.
Technically, Excel’s Solver may not put exactly integers in the changing cells. But if not, they will be fractions which are very, very close to being integers, such as 0.9999999.

When certain variables are constrained to be integers, the model is referred to as an integer programming (IP) model. Compared to models without the integer constraint, there are five things you should know about IP models:

1. Larger models which constrain changing cells to be integers can take **much longer to solve**. For large problems, Solver usually has to employ special algorithms. These are not as efficient as the Simplex LP algorithm used on models which can accept fractional values.
   - Try not to use integer constraints unless you absolutely have to.

2. While a non-integer version of a model may have a solution, the same model with the integer constraint **may not have any feasible solution** at all.
   - Example: Place three people into two chairs. If Solver is allowed to use fractions of people, there’s a solution. If the number of people per chair must be an integer, however, then no solution exists which places all three.

3. It’s tempting to remove the integer constraint, solve with fractional values, and then just **round off the fractional values to integers. Not a good strategy.** The non-integer solution sometimes is **quite different** from an integer solution, so rounding the fractions won’t give the correct answer.

4. Solver’s **sensitivity report is not available** for IP models. Its calculations are not meaningful when the model uses integer constraints. With an IP model in Excel, sensitivity analysis must be performed manually. Enter different values for the RHS constraint numbers and rerun Solver.

5. Most versions of the Excel Solver have an option to “Ignore Integer Constraints,” and it’s turned on by default. For all these models, you must go into the Options menu and **uncheck** that option. Otherwise, that option overrides the “int” constraint, and Solver can put fractional values into cells that are supposed to be integers.

**Notes on Binary IP models**

In business decision-making, a quite useful form of IP model is the **Binary** IP model. In a Binary IP model (BIP), some of the changing cell decision variables are constrained to be only 0 or 1. No other value is allowed. These 0-1 binary variables allow you to model “either-or” and “yes-or-no” situations in a linear model. If models are kept linear, they are much easier to solve.
In Solver’s constraint dialog box drop-down menu up to now, we usually have been selecting the inequality signs. But that same drop-down menu offers an option to constrain a variable to be binary (“bin”).

When constraining certain changing cells to be binary, it’s also necessary to go into Solver’s Options menu and uncheck the option that says, “Ignore Integer Constraints.”

**Contingent decisions in Binary IP models**

An example of how binary variables can be used is a decision where fixed costs are involved. This is known as a contingent decision. If a certain choice is made, then the fixed cost will be incurred. If that choice is not made, then its fixed cost is not incurred. The fixed cost is contingent on the choice. It is required to be included if that choice is selected, and otherwise it is zero.

That either-or situation with the fixed cost would appear to make the model nonlinear. It’s a lump sum that may or may not be added on. By using a binary 0-1 variable, though, a contingent fixed cost can be included in a model. And it’s still linear.

The technique is this. Simply create a changing cell for the yes-or-no decision on that option. Constrain that changing cell to be binary: 1 if the option is chosen, and 0 if it is not. Then multiply your cost computations, including any fixed cost cell, by that binary variable. The result is that if that option is selected, the fixed cost is included in the sum of the costs.

To illustrate, let’s assume that one option in a possible solution to a problem is to use Machine X to produce the desired items. If we use Machine X, there is a $2,000 setup cost. If we don’t use Machine X in the solution, of course, the setup cost for Machine X is zero.

In the model, we’ll add a binary changing cell. Then, one term in the total cost will be: Binary Changing Cell × $2,000. And in computing how many items will be produced, one of the terms will be: Binary Changing Cell × Number of Items Produced on Machine X.

Those two terms accomplish what we want, with simple linear multiplication formulas. If any items are to be produced on Machine X, then the Binary Changing Cell must be set to 1 (so the second term can be a positive number of items). And if the Binary Changing Cell is set to 1, then the $2,000 gets included in the total cost.
Mutually exclusive decisions in binary IP models

Binary variables can model a mutually exclusive decision, too. A mutually exclusive decision is one where there are several options, but only one can be chosen. Choosing one automatically excludes all the others.

In other words: If A is chosen, then B and C cannot also be chosen. Choosing A rules out B and C. Similarly, choosing B rules out A and C.

To model a mutually exclusive decision, place a binary yes-or-no decision variable (i.e., a binary changing cell) on each choice. Multiply the factors in the objective function by these binary variables. Then constrain the sum of these binary variables to be equal to 1. Thus, only one of the mutually exclusive options will be selected in the solution.

Notes on the examples and practice problems

Setup or startup costs and units produced

At the end of Chapter 7, the Hillier book uses an extension of the Wyndor example to show how binary variables can work. There is a setup cost for doors and a setup cost for windows. The setup cost for a product is a fixed cost of producing that product. It is incurred only if any units of that product are actually produced. Therefore, the model must be constructed such that if any units of a product are produced, then the setup cost is included in the objective function. If zero units of a product are produced, then the setup cost for that product is not added into the objective function.

This if-then situation is handled by binary variables. There's a binary variable for each product. Those binary variables are multiplied by the amount of each setup cost in the objective function. And the binary variables also are multiplied by the number of doors and windows. So if any doors or windows are to be produced (binary variable =1), then the amount of the setup cost is included. If one of the products will have zero units produced (binary variable = 0), then no setup cost will be added, since the setup cost is multiplied in this case by zero.

Electric Generator practice problem. This problem is posted on Canvas, along with a detailed explanation and a short video showing how to build the model. It’s another problem with a fixed startup cost.

There are two sets of changing cells. One is for the number of megawatts produced by each generator. The other is the binary yes-or-no decision variable for the startup cost. The binary changing cells are used in two ways.
First, the objective function is total cost to generate the required megawatts. That is the sum of the unit costs plus any startup costs. The 0-1 binary changing cell gets **multiplied by the startup cost** for each generator and added to the objective function. If the decision is yes, use that generator (0-1 variable = 1), then the startup cost gets added in. If the decision is no, do not use that generator (0-1 variable = 0), then the startup cost is zero.

The explanation on the Canvas page also describes how to set the capacity constraints in a clever way, so they use the binary variables, too.

In this formulation, then, anytime a generator is used, two things happen. First, the startup cost is included. Second, the megawatts produced can be a non-zero number (multiplied by 1). Anytime a generator is NOT selected, then startup cost will come out to zero, and megawatts produced will come out to zero (multiplied by a binary of zero).

It’s also possible that if there is a constraint on total units, then the binary variable will be multiplied by that constraint value. And total units produced will be bounded by exactly that number.

**Airplanes practice problem.** This problem is posted on Canvas. The company must decide which of three airplane models to produce. Each model has an associated fixed setup cost. Whichever model or models are put into production, those setup costs must be included.

Note how the 0-1 “Produce Any?” variable is used. First, multiply that binary variable by the startup cost, and the model includes the startup cost only for the selected customers.

Next, set a constraint on the number of airplanes produced for each customer. Multiply the 0-1 “Produce Any?” variable by the maximum possible number of units per customer. Constrain the units to be less than or equal to that multiplication. That way, if the customer is not chosen, then the maximum number of airplanes will be zero. If the customer is chosen, then production will be less than or equal to the maximum for that customer.

Thus, the model will apply the marginal revenue only if airplanes are produced for that customer – which means Solver sets the binary variable to 1.

**Sales Rep Assignment problem.** This problem is posted on Canvas. It’s not from the Hillier book, but it’s an excellent practice problem because it’s a little more complicated. It features a contingent fixed cost with some additional variables. It
illustrates how the binary variable must be used in the capacity constraint to set an “upper bound.”

**California Manufacturing Company example in Hillier.** A practice spreadsheet for this problem also is posted on Canvas. You can formulate and solve the model on the blank sheet before you look at the solution worksheet.

This is a problem with a **mutually exclusive decision**. The challenge with a mutually exclusive decision is that it's another if-then issue. If one of the options is chosen, then the others cannot be. Once again, inserting an if-then constraint would make the model nonlinear.

A binary variable can be used, though, to establish that constraint and still keep the model linear.

Here’s the problem. The company plans to build one or possibly two new factories, and perhaps also a warehouse. It is considering either Los Angeles or San Francisco. But it needs at most just one warehouse, so if it chooses one of those two cities for the warehouse, the other is automatically eliminated. Only one city can be chosen.

The four choices are:

1) Factory in L.A.
2) Factory in San Francisco
3) Warehouse in L.A.
4) Warehouse in San Francisco

Choices 3 and 4 are mutually exclusive.

The estimated NPV for each of the four possible choices has been determined. And there’s a binary variable for each warehouse choice. This binary variable is multiplied by the NPV and the capital requirement for each warehouse, to calculate the total NPV and the total capital.

To make them mutually exclusive, we simply add a constraint that the sum of the two binary variables for Choice 3 and Choice 4 must be less than or equal to 1. So only one can be chosen.

Another way to see what’s going on in that spreadsheet is to change the capital requirement for the warehouse in Los Angeles from 5 to 4. Rerun Solver. You’ll find that the optimal solution now has only one factory, located in L.A., and one warehouse, also located in L.A.