Module 5: Extending Our Modeling Ability

Reading Material: 5.1 – Introduction

To this point, we have set up and solved just one basic LP model and analyzed it extensively (some might say we studied it to death!). The present module extends our modeling repertoire while we are still solving models graphically. Specifically, we will look at proportion or ratio constraints, constraints that are greater than or equal to, and also examine a decision scenario in which the objective is to minimize a criterion.

Upon module completion, we will have mastered the foundations of model development, which will allow us to develop and solve much more realistic size problems and additional realistic, practical circumstances, fully taking advantage of linear programming's power as a decision tool.

This module will use the same LEGO production backdrop and explore two additional scenarios that augment or alter the previous scenario. In the first new scenario, our goal is to maximize sales in the production of tables and chairs as constrained by the original three resources (small LEGO blocks, large LEGO blocks and Capacity) and an additional proportion constraint. This constraint, for example, could be viewed as a restriction related to demand for our furniture, a specific company policy constraint, or something along these lines.

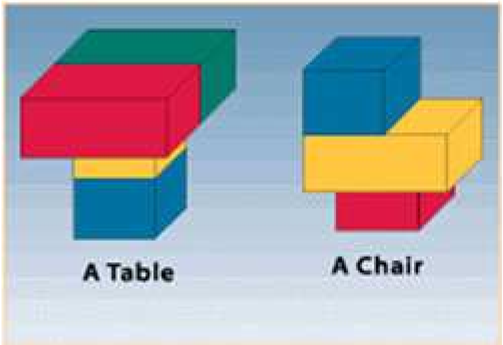
In the second scenario, while still producing LEGO tables and chairs, we re-shuffle problem requirements substantially and recast it as a "minimize cost" problem. Here, we use greater than or equal to constraints for the first time and experience our first minimization problem.

Reading Material: 5.2 – Proportion/Ratio Constraints – Keep the Linear in Linear Programming

Here is the original problem description of the first LP model that we solved previously in Module 3.

5.2.1 EXAMPLE: Producing LEGO Furniture

Consider this example: A company produces tables and chairs from LEGO blocks. A table is made up of two large LEGO blocks and two small LEGO blocks. A chair is made up of one large and two small LEGO blocks.



Tables are sold for $16 apiece, and chairs are sold for $9 apiece. Space is limited, and the tables and chairs are measured in terms of cubic units (cu units) – each table takes up six cu units of space, and each chair four cu units. Also, there are 300 large LEGO blocks and 400 small LEGO blocks available for use in furniture production. Determine the production of tables and chairs that maximizes sales.

Here is the graphical depiction of our feasible region for the same problem.

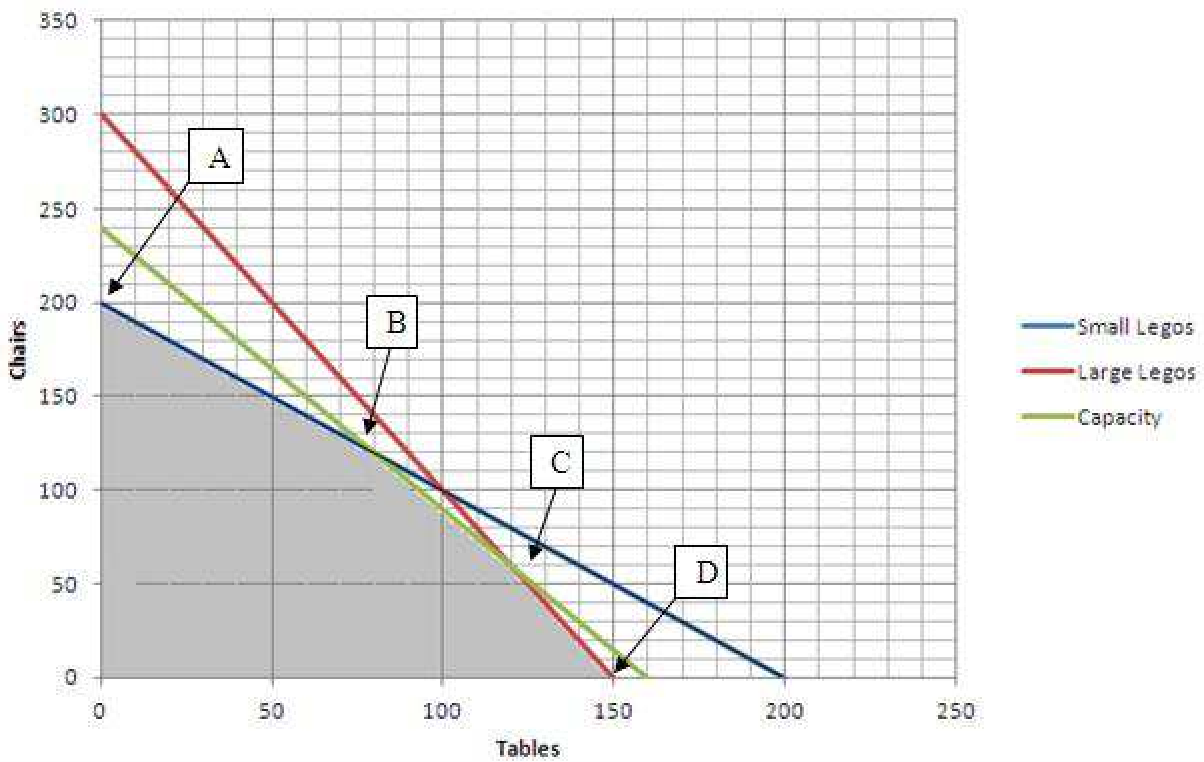


Figure 5.1

5.2.2 The Additional Ratio Constraint

Add the following furniture composition constraint to the model: Because of higher demand, the number of tables in the optimal production mix must be at least 80% of the total amount of furniture produced.

This constraint can be constructed in a number of different ways. For instance, one might think of this as a ratio:

Tables/(Tables + Chairs) >= .8

This would translate to "the ratio of tables to total production (tables + chairs) must be at least .8 (80%)."

THIS IS AN EXAMPLE OF A NON-LINEAR EXPRESSION. Not yelling, just emphasizing!

As we've mentioned, LP is called linear for a reason. Representing the constraint in this fashion (non-linear) is not appropriate for our use.

However, with just a little algebraic magic, we can transform that ratio into an equivalent linear expression. Multiply both sides by the expression (Tables + Chairs). This makes the equation become:

Tables >= .8(Tables + Chairs).

Actually, this way of formulating the constraint initially might be more intuitive to the modeler that the previous form. If you translate the constraint written this way back into words, it says "The number of tables must be greater than or equal to 80% of the total production." This expression is in fact linear. It doesn't look like our other previously created constraints (yet), but it is a linear expression of our proportion/ratio restrictions.

We can further manipulate this constraint in a variety of useful ways. First, applying the distributive property, we get the following equivalent expression:

Tables >= .8 Tables + .8 Chairs.

Ok, not so exciting – and I'm not sure if that was really the distributive property or the associative property, but it sure sounded impressive! I'm sure you noticed that the other three constraints in the original LEGO production model had all decision variables on the left-hand side of the inequality and a number on the right hand side of the inequality. Could we further rearrange terms in this constraint to match that pattern? Yes, by simply moving the terms on the right-hand side to the left (by subtracting –.8Tables – .8 chairs from both sides) we get:

.2Tables – .8 Chairs >= 0.

It looks a little odd (a negative coefficient for chairs), but it is correct and is in what historically has been termed *standard form*, and what we will call “row/column” form (in preparation for using EXCEL).

But, for finding solutions graphically, there might be an even better way for representing this constraint! Move the chairs back to the other side of the inequality – don’t scratch the floor, though.

.2 Tables >= .8 Chairs

Further, divide both sides by (.2) and we get a nice relationship between production of tables and production of chairs.

Tables >= 4 Chairs.

Thus, to satisfy the production restriction that we wanted to add (80% of all production must be tables), we will add a constraint graphically that forces the number of tables to be at least 4 times the number of chairs.

For me, this is the most useful form of the constraint while we are solving LP models graphically. And that is what we will do in the next section.

5.2.3 Graphing a Ratio Constraint

In terms of adding this fourth constraint to our graph, any of the linear forms of the constraint we previously derived are “graphable.” Feel free to differ from my approach below if you have adopted another equivalent approach to solving these models graphically. There is nothing wrong with alternative, equivalent approaches.

I will be using the Tables >= 4 Chairs representation as my additional constraint in constructing the graph. As before, we will first plot the line Tables = 4 Chairs. Then, we will determine which side of the line is feasible (exactly the process employed in Module 3).

The line Tables = 4 Chairs passes through the origin (0,0) and travels to the “northeast” (using our graphing compass again). You might recognize some sort of slope/intercept thing going on here. The line moves to the northeast by increasing the values of tables 4 units for every 1 unit increase of chairs. We could also simply pick a few random points that fall on the line, and then connect the dots (I always liked those kinds of puzzles). For instance, when Chairs = 25, Tables = 100. When Chairs = 50, Tables = 200, and so on. Figure 5.2 shows the placement of the constraint.

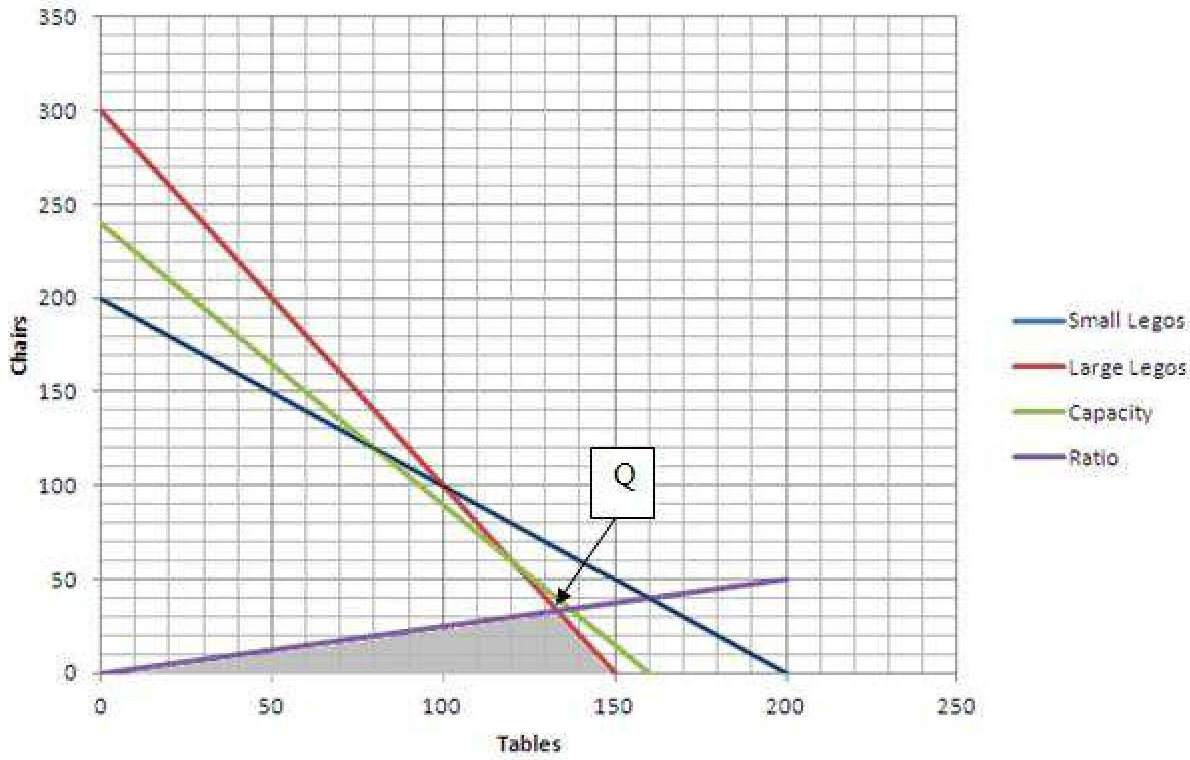


Figure 5.2

Now, which side of the line is feasible? Again, I would simply pick two random points, one on each side of the line, and see which one satisfies the original constraint (Tables >= 4 Chairs). Let's pick 100 Tables and 0 Chairs, and 100 Chairs and 0 Tables. Which of those two points satisfies the constraint? Yes, the point 100 Tables, 0 Chairs, which means the feasible region lies to the southeast (down and to the right) of the line we just plotted.

The result – our feasible region after adding this fourth constraint is a triangle. The extreme points are the origin (0,0), which we can ignore, the intersection of the large LEGO constraint and the Table axis (150,0) and the intersection of the newly plotted ratio constraint and the Large LEGO constraint. As before, we need to find the production amounts of tables and chairs that this new extreme point represents and compare it to the objective function value of (150,0).

So, the intersection point (Point Q) is found where these two equations intersect:

TAB = 4 CHR

2TAB + CHR = 300

By substituting 4CHR for TAB, the second equation becomes 2 * (4CHR) + CHR = 300.

This simplifies to 9CHR = 300, or CHR = 100/3 = 33.33.

Back substituting into the first equation, we see that TAB = 4 * (100/3) = 400/3 = 133.33.

We have previously briefly mentioned the possibility of fractional decision variable values. As you can see, they occur. LP in its basic form allows fractional values. In our present production scenario, it doesn't necessarily make sense to consider producing fractional tables and chairs. But for right now, and until we talk more about integer requirements in a section aptly called Integer Programming, we are just not going to let fractional values worry us.

Why don't we worry about them now? Two main reasons: (1) I'd prefer to postpone computational complexity issues associated with adding integer requirements until later, when we are more comfortable with LP modeling in general. (2) When we force decision variables to be whole numbers, we lose the ability to see sensitivity analysis (at least classic sensitivity analysis). Now, in practice, if we really need integer requirements, we really need integer requirements!! But we are methodically building a foundation of knowledge in this first set of modules, and I don't want complexity to distract us. We will get to complicated modeling soon enough.

Back to finding the optimal production mix: To find the optimal production mix, we need to compare the objective function value of (150,0) to (133.33, 33.33). (Do we see why we do not have to evaluate the origin, producing no tables and no chairs?). Recall that each table adds $16 to our sales and each chair $9, so the two solutions provide $2400 (150 * 16 + 0 * 9) and $2433.33 (133.33 * 16 + 13.33 * 9), respectively. So, by adding the 80% requirement to our previous production model, we find that the optimal solution is to produce 133.33 tables and 33.33 chairs.

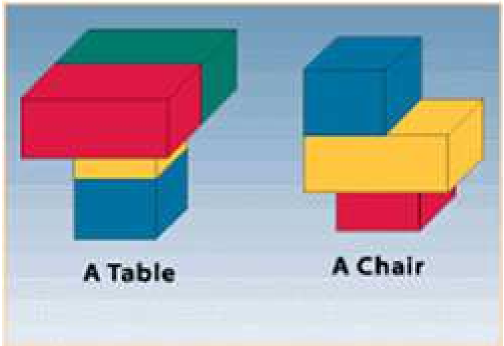
Side Note: The optimal solution we determined in Module 3, before adding the 80% requirement, did not satisfy this requirement. Thus, the original determined optimal mix of tables and chairs was not even a candidate extreme point for our revised problem and the optimal solution was different.

In the next section, we re-craft the LEGO furniture production problem so we can see different kinds of constraints within the context of a minimization problem.

Reading Material: 5.3 – Minimization Problem – LEGO furniture Version #2

Here is the re-casted LEGO production situation:

Using the specifications below, find the least cost solution in the production of tables and chairs. A company produces tables and chairs from LEGO blocks. A table is made up of two large LEGO blocks and 2 small LEGO blocks. A chair is made up of 1 large and 2 small LEGO blocks.



It costs $16 to make a table and $10 to make a chair. Space is limited, so we can make no more than 175 total pieces of furniture. For variety's sake, we must make at least 60 chairs, and at least 60 tables. Our production plan must use *at least* 220 large LEGO blocks and *at least* 300 small LEGO blocks. Find the minimum cost production levels of tables and chairs that meet these requirements.

We will now set up the algebraic linear programming model, moving a little quicker than we did previously.

Step 1: Decision Variables – same as previously. TAB = number of tables produced, CHR = number of chairs produced.

Step 2: Objective: Our goal here is to minimize cost, and algebraically cost is $16TAB + $10 CHR.

Step 3: Constraints. The context is different than the previous version of the problem, but the tables and chairs use the same number and type of LEGO blocks as before. There are five constraints: Total production (no more than 175 tables and chairs combined), individual chair production (at least 60), and individual table production (at least 60), large LEGO block usage (at least 220) and small LEGO block usage (at least 300).

Algebraically, this equates to the following five constraints (in the same order as mentioned above).

TAB + CHR <= 175

CHR >= 60

TAB >= 60

2TAB + CHR >= 220

2TAB + 2CHR >= 300

Note how the individual production of tables and chairs constraints are two separate equations.

We have created our LP model and will now solve it graphically.

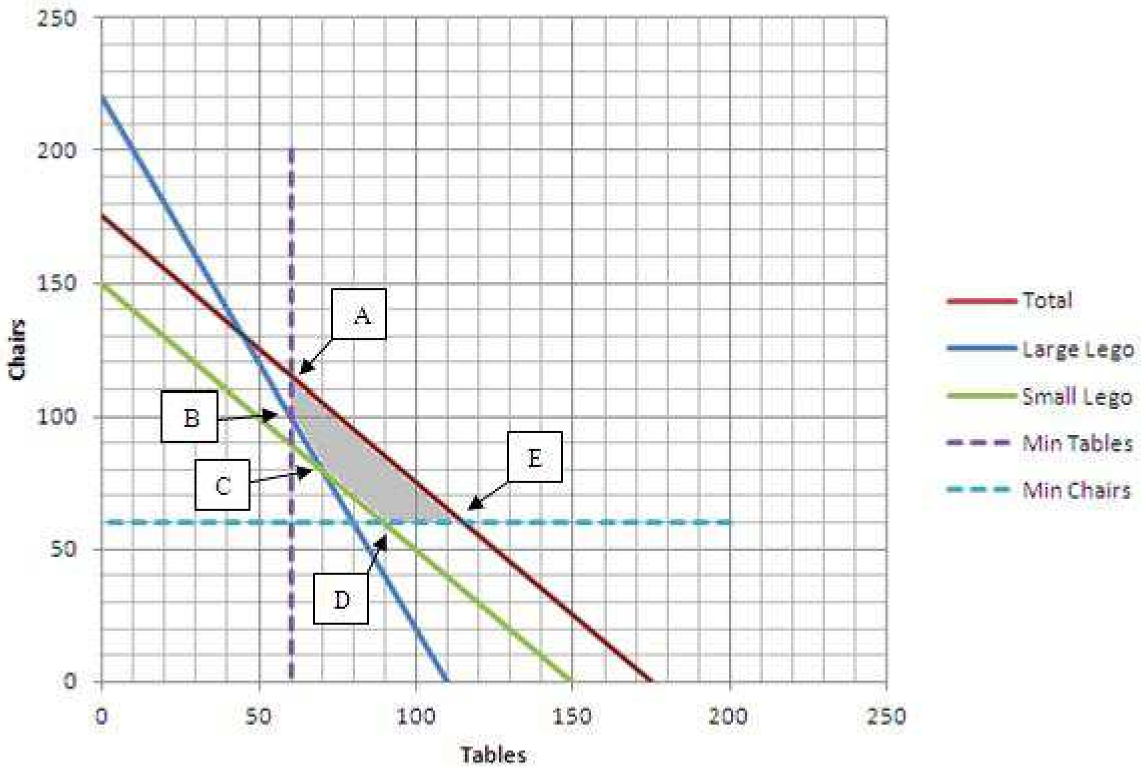
Finding a solution graphically uses the same basic process regardless of whether is a MIN or MAX objective. As before, for each constraint, we will find the point where the constraint intersects each axis, then connect the dots, and determine which side of the resulting line satisfies the inequality portion of the constraint (feasible area).

The capacity constraint intersects each axis at 175 (0,175) and (175,0) and the feasible area falls on the line and back toward the origin.

The next two constraints are unique – CHR >= 60 is actually a horizontal line, where the feasible region lies on the line and up. This represents the requirement that we must produce at least 60 chairs. Similarly, TAB >=60 is a vertical line (at 60) whereby all points on the line and to the right (outward) are feasible. This represents the requirement that our solution must include at least 60 tables.

The final two constraints, minimum usage of large and small LEGO blocks, are also both greater than or equal constraints, so when we plot the line, all points on the line and outward from the line are feasible. The Large LEGO line intersects the TAB axis at (0,110) and the CHR axis at (220,0), while the Small LEGO line intersects the TAB axis at (0,150) and the CHR axis at (150,0).

Figure 5.3 shows the feasible area, with extreme points notated. As before, using the old Wilson brute force method, we will in turn evaluate each extreme point and the point with the smallest objective function value (because we wish to minimize cost) is the optimal solution



Point	Equation		TABLES	CHAIRS	Cost
A	TAB=60	Min T	60	115	2110
	TAB + CHR = 175	Space			
B	TAB=60	Min T	60	100	1960
	2TAB + CHR = 220	Large			
C	2TAB + CHR = 220	Large	70	80	1920
	2TAB + 2CHR = 300	Small			
D	CHR = 60	Min C	90	60	2040
	2TAB + 2CHR = 300	Small			
E	CHR = 60	Min C	115	60	2440
	TAB + CHR = 175	Space			

Figure 5.3

From the graph, we have identified five extreme points (A thru E). We evaluate all five, even though some of you may have assessed that two of the extreme points (A and E) cannot be optimal. Because costs are a positive value, and our objective function is linear, points B and D will always be superior (smaller total cost) to A and E. If this minor point is not clear, don't worry; we'll move on just evaluating each extreme point anyway.

The table below the graph identifies the two constraints that intersect at each extreme point. Also listed is the number of tables and chairs produced that this extreme point represents, and the overall cost (the objective function value) for the production level.

All extreme points except Point C involve either the minimum production requirements of tables or the minimum production requirement of chairs. This makes solving the system of linear equations fairly easy at these points. This minimum requirement of 60 can be directly substituted into the other constraint, and the production levels almost immediately determined with a minimum of algebra required.

For Point C, we can use an algebraic shortcut to determine the numbers of tables and chairs it represents. Note how similar the two equations are:

2TAB + 2CHR = 300

2TAB + CHR = 220

We can actually subtract the second equation from the first to solve for the value of CHR. We line up like terms and subtract. Thus, the TAB decision variable goes away (2 – 2 = 0), the CHR decision variable is left with a 1 coefficient (2 – 1 = 1), and the right-hand size becomes 80 (300 – 220). Thus, we have solved for CHR = 80! We can then back substitute that into either of the two equations, and we find that TAB = 140/2 = 70.

This subtraction process simply consolidates numerous algebraic steps. Again, if you need to go through the entire algebraic process to get to the solution, that is certainly fine. My showing this was just to give you a flavor for other simplifying approaches that exist in determining intersection points using linear algebra (and, admittedly, I was growing weary of typing out the entire process!).

As we refer back to the table, we see that Extreme Point C turns out to be the optimal production mix of tables and chairs for our restated LEGO production model. The optimal solution is to produce 70 tables and 80 chairs for a cost of $1,920.

Now that we have found the optimal solution, we could do a similar analysis to what was done in Module 4 with the original problem, identifying binding constraints, the robustness of the optimal solution, the marginal value of some of our constraints, etc. We won't, because Module 4 was only for illustrative purposes if future Sensitivity Analysis topics (see Module 7) need visual clarification. I mention this to make sure it is clear that minimization problems and problems with multiple types of constraints can also take advantage of the automatically generated Sensitivity Analysis reports to gain even more insight about the problem under study.

Also, here at the end of the module, I want to mention that we have not yet dealt with an equal-to constraint. In some ways, they are simpler to 'plot' than inequalities, because the feasible area that relates to such a constraint is in fact just a line! We will have to deal with equality constraints soon enough in many domains, and if you are faced with one in two dimensions, just remember how such a constraint affects determining feasible regions (a line segment is feasible, not an area).

Reading Material: 5.4 – Summary

We have now covered a major introductory foundational portion of the class. We have constructed some LP models, saw a variety of constraints, and are getting comfortable with the algebra needed to successfully master LP modeling.

It is not particularly practical to solve LP models by hand; for one thing, that means a problem only has two decision variables. But history has shown over my classes that by spending this time and effort with our algebra fundamentals, we receive a good foundation that allows us to quickly move beyond the simple "toy" problems, and develop increasingly more practical modeling scenarios. Often, by the time we hit the sixth or seventh week of the semester, your colleagues have started to develop models on their desktops for problems they face day in and day out – scheduling their staff, managing their supply chain, assigning umpires to their little leagues games, all kinds of things. It really is worth the time, not just because I learned LP before we had electricity and had to do problems by hand.

There are a few more practice problems that follow this module, but then we'll be ready to move on to using the Solver in EXCEL to create LP models. We have to walk before we can run, and be ready – the pace will soon quickly pick up speed!

Over the last few modules, I have interspersed some other fundamentals that are worth restating and summarizing before we move forward.

LP requires linear relationships. There is a field of study called non-linear programming. Solutions are never guaranteed in that setting, because non-linear solution algorithms are just intelligent, brute-force guessing algorithms. Non-linear modeling is beyond the scope of this text – but certainly can add value to the modeler.

Linear programming requires a single, quantifiable objective (Max sales, min cost, etc.). We will look at variations to this toward the end of our text – because LP is algebra on steroids, then goal programming is LP on steroids – allowing us to explicitly deal with the realistic setting of having multiple, conflicting objectives.

Linear programming in its pure sense allows us to have fractional solution values. Integer programming will allow us to force decision variables to have whole numbers or even binary (yes/no) decisions should that be appropriate.

Hopefully, we have received a taste of some of the basic modeling skills needed to be successful in practice, as well as ways of gaining additional insight in a problem under analysis. We are ready to tackle more realistic-sized problems. EXCEL, here we come!