Module 3: Introduction to Linear Programming

Reading Material: 3.1 – Introduction

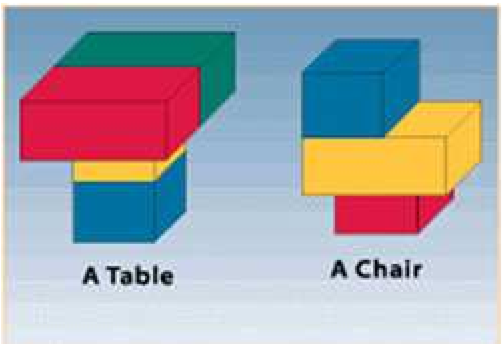
In this module, we will create our first linear programming (LP) model and solve it using pictures – actually, a combined algebraic/graphical approach. In creating and solving our first LP model, we illustrate perhaps the main overarching theme of problem solving that runs throughout the text. The steps of this theme, this problem-solving process are:

1. Identify what we control or what we want a model to help us make decisions about.
2. Identify our goal or objective(s) in determining the best decisions in Step 1 above.
3. Identify the items that constrain us from achieving even greater levels of our goal as we try to find the best solutions to those things we control.

As discussed in the previous module, this philosophy is the essence of the class and epitomized by the quantitative models we will be developing throughout.

Next is our kickoff problem, modified from a scenario first developed by Pendergrast (1997). Normally, I let my class grapple with this as an in-class exercise, seeing if they can conceptually leap from toy problems (pun intended?) worrying about how many coins we have in pocket change or how many people went on our field trip (the fun problems we used in Module 1) to more realistic business settings (like we're going to be managing LEGO furniture production, right?!). Struggling is okay, as the proverbial "A-HA" moment is always best when you experience it yourself!

Reading Material: 3.2 – 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 6 cu units of space and each chair 4 cu units. The produced furniture can take up a maximum of 960 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.

3.2.1 – Linear Programming Model Definition – Set-Up

When analyzing decision problems in this book, with few exceptions, we should start with the three step process mentioned in 3.1. This systematic approach will serve us well in creating mathematical models to assist analyzing decision scenarios.

1. What do we control or what do we want a model to help us make decisions about in this scenario? Hopefully, the answer is somewhat clear – the number of LEGO tables to make (TAB) and the number of LEGO chairs to make (CHR). The three-character abbreviation is the algebraic representation of our *decision variables*.
2. What do we want to accomplish? We would like to maximize sales in the production of LEGO furniture. Because we are creating a mathematical representation of our business situation, the *objective function* must be a quantifiable statement. In the problem description, we are given the sales per unit for tables and chairs. If we took the values as "profit," one would assume that detailed accounting analysis, market factors, and other pertinent facts were used to help derive these values. Mathematically, then, our objective function is: MAX 16 TAB + 9 CHR.
3. What constrains us in producing tables? We have a limited amount of SMALL LEGO BLOCKS, LARGE LEGO BLOCKS and the CAPACITY of furniture (as measured in cu units). We must also specify these constraints mathematically.

How are small LEGO blocks used in the production of furniture? Each table uses two small LEGO blocks, as does each Chair. There are 400 total available to be used in furniture production. Algebraically:

2 TAB + 2 CHR <= 400.

This states that the total number of small LEGO blocks used in making tables (2*TAB) added to the total number of small LEGO blocks used in making chairs (2*CHR) cannot exceed (<=) 400, the available supply.

Similarly, consider large LEGO blocks. Tables use two large LEGO blocks, chairs only one. Thus,

2TAB + 1CHR <= 300.

As above, the total number of large LEGO blocks used in making tables (2*TAB) added to the total number of small LEGO blocks used in making chairs (1*CHR) cannot exceed (<=) 300, the available supply.

Capacity of the furniture – each table takes up 6 cu units and each chair 4 cu units, and we have a total area of 960 cu units to store produced furniture. The constraint:

6TAB + 4 CHR <= 960.

Combining these parts into a standard algebraic formulation, we have our first linear programming model, which describes our LEGO production situation.

MAX	16 TAB +9 CHR
ST	2 TAB + 2 CHR < = 400
	2 TAB + CHR <=300
	6 TAB+ 4 CHR <=960
	TAB, CHR >=0

3.2.2 Model Definition: Comments

What is “ST” and then that odd constraint at the bottom?

“ST” is normally seen as “s.t.” or “such that” or similar verbiage. It translates to “Here come the constraints.” It is a habit of mine and is seen in most books.

The odd-looking constraint at the bottom of the model specification (TAB, CHR >= 0) is another formalism traditionally found in an algebraic representation of a mathematical programming model. Specifically, this is a non-negativity constraint. It is a statement indicating that the decision variables can only be 0 or greater (i.e., non-negative). Seems reasonable, correct? It would be illogical to allow a model to find an optimal solution that suggested making –5 tables!

Looking ahead, in some versions of EXCEL, one must explicitly tell the SOLVER (the add-in we will be using to solve linear programming models) to force all non-negative solution values. If one forgets, some pretty wild model solutions can result (indicating the model has blown up). There may be times when one would want negative values for a decision variable, but it is very uncommon.

Another point for future topics, the formal statement of the LP model variables is also used to indicate other decision variable requirements. These requirements could be that decision variable values should only be whole numbers (integers) or binary values (0 or 1).

For now, we are not concerned about solutions with whole numbers– even though it would not make sense to produce fractional tables and chairs. We will address forcing decision variables to be whole numbers later in the book. Do not let fractional values worry you at present.

Another point worth mentioning is that LP models assume the modeler specifies one objective to be optimized. As we all know, sometimes reality confronts us with multiple, conflicting objectives. Later in the text, we relax this single-objective assumption and consider explicit multiple objective approaches to modeling (goal programming is one such technique), To summarize, we have created a linear programming model to analyze our LEGO furniture production scenario. Our LP model has two decision variables and three constraints, with a single objective of maximizing sales. The basic LP model assumes that our decision variables cannot be negative and are allowed to have fractional values. The model is linear because all the relationships are linear.

Also, note that this initial problem has only less than or equal constraints. In general, LP models will have three types of constraints – <=, >= and =. This problem did not have any stated requirements such as "Must make at least two tables." This requirement would result in a greater than or equal to constraint (TAB >= 2). Module 5 will show us some extended LEGO production examples to illustrate a wider variety of constraints.

We next turn our attention to how we can use this model to find the optimal production levels of tables and chairs. With our (reinvigorated) knowledge of algebra and the help of pictures and mathematical brute force, we will find the optimal production levels of tables and chairs that maximize sales. Sharpen your pencils, unleash your knowledge of the slope intercept form of a line, and let's solve this LEGO production linear programming model!

Reading Material: 3.3 – Graphical Solution Process for LP Models – Overview

So I was being facetious. I don't know if that translates well in book form.

Actually, sharpening pencils may be appropriate, but we really don't need to recall the slope intercept form a line (that's y = mx + b for those of you keeping score at home). If you do recall it and can benefit by using it, that is wonderful. We are going to illustrate a solution methodology that is less elegant but will still get us to the same point – *the optimal point*!

In a step-by-step process, here is how we will solve this LP model –

1. Plot all constraints on an x-y graph. The x-y graph represents the non-negative production levels of the tables and chairs and in general represents the possible solution values of decision variables.
2. Find the feasible area/region that represents possible solutions that satisfy the intersection of all relevant constraints.
3. Identify the extreme points of the feasible area (feasible region) – those where two lines intersect to form the boundary of the feasible region. One (or more) of the extreme points WILL BE the optimal solution of the LP model – which, in our first case, represents the optimal level of producing tables and chairs to maximize sales.

We will go through this process methodically step-by-step.

3.3.1 PLOT Model CONSTRAINTS

Small LEGO blocks will serve as our first example on graphing constraints.

2 TAB + 2 CHR <= 400

First, using the non-elegant, non-slope-intercept method, we are first going to plot the LINE that represents the constraint on our graph, then worry about which way is feasible later (the inequality portion). So, we are going to plot 2 TAB + 2 CHR = 400.

Our graph will have a horizontal axis and a vertical axis representing levels of production for the furniture. Arbitrarily, let us choose tables to be represented on the horizontal (X) axis, and chairs to be associated with the vertical (Y) axis.

Consider this – if we did not produce any tables (TAB = 0), how many CHRs could we produce with 400 small LEGO blocks? (Answer = 400/2 = 200).

Likewise, if we did not produce any chairs (CHR = 0), the constraint equation tells us that we can produce how many tables? (Answer = 400/2 = 200).

Therefore, the last two statements identified the intersection points of the Y-axis (TAB = 0) and the X-axis (CHR = 0) for the small LEGO constraint.

We can then plot the line form of this constraint by making a point at (0,200) – representing producing 0 TAB and 200 CHR – and a point at (200,0) – producing 200 TAB and 0 CHR – and then connecting the points with a line (like connecting the dots). The resulting line represents the infinite number of solutions in which all 400 units of small LEGO blocks are used in producing tables and chairs.

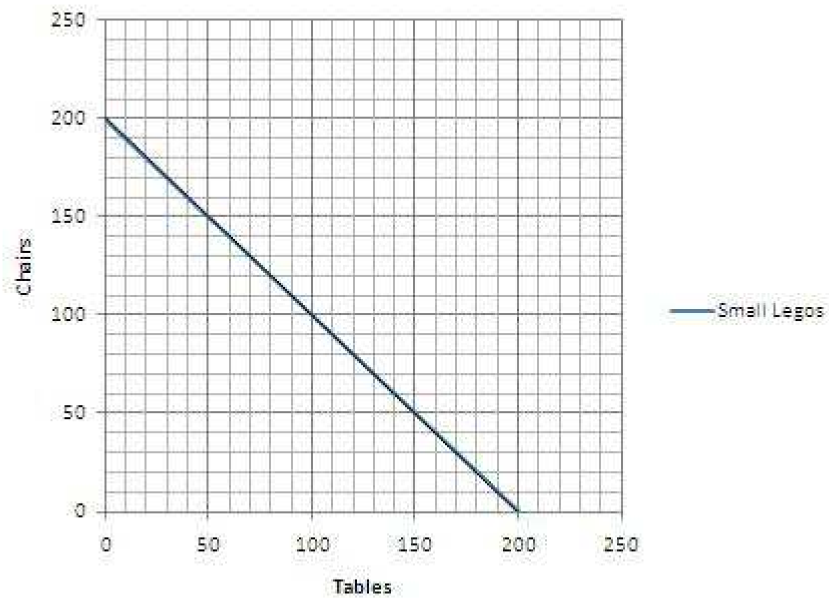


Figure 3.1 shows the small LEGO constraint graphed.

One more item to consider – which side of the line is “feasible”? Well, we have a maximum of 400 small LEGO blocks that can be used, but we certainly can use fewer than 400. So, feasible combinations of producing TAB and CHRs can be found on the line just drawn and BACK TOWARD THE ORIGIN of the graph. The shaded area in Figure 3.1 shows this.

We repeat the process for the next two constraints. In some ways, the two steps described earlier are done simultaneously. We plot a constraint, determine where its feasibility lies, and then look for the intersection with other constraints already plotted on the graph. Note that as we add constraints, the feasible area can only get smaller, because we are placing more and more restrictions on possible solutions.

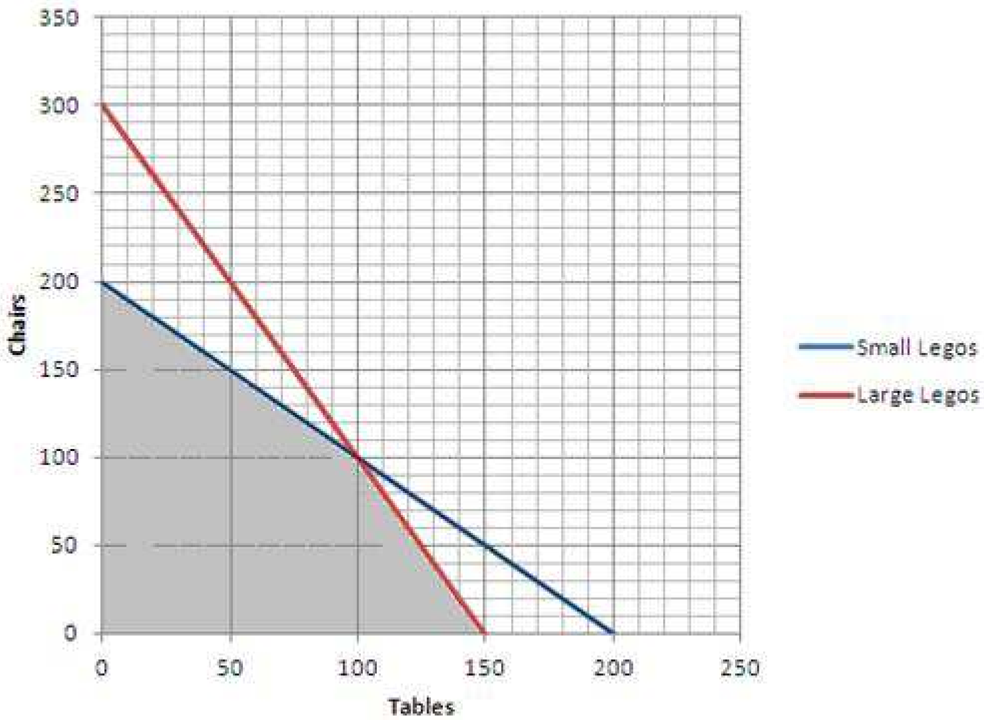


Figure 3.2 shows the addition of the large LEGO constraint. Again, using our brute force approach, we set one variable to 0 and find the value of the other variable, representing an axis intersection point. We then reverse the roles of the decision variables and find the other axis intersection point. Connect the dots, and we have another constraint plotted on the graph.

For this constraint (2TAB + CHR <= 300), when we set TAB = 0, CHR = 300 (0,300). When CHR is set equal to 0, TAB = 150 (150,0). Those are the two axis intersection points, and the line between them represents all production levels at which exactly 300 large LEGO blocks would be used. Because we can use fewer than 300, the feasible area again lies back toward the origin, and we see our feasible area after two constraints have been pared down.

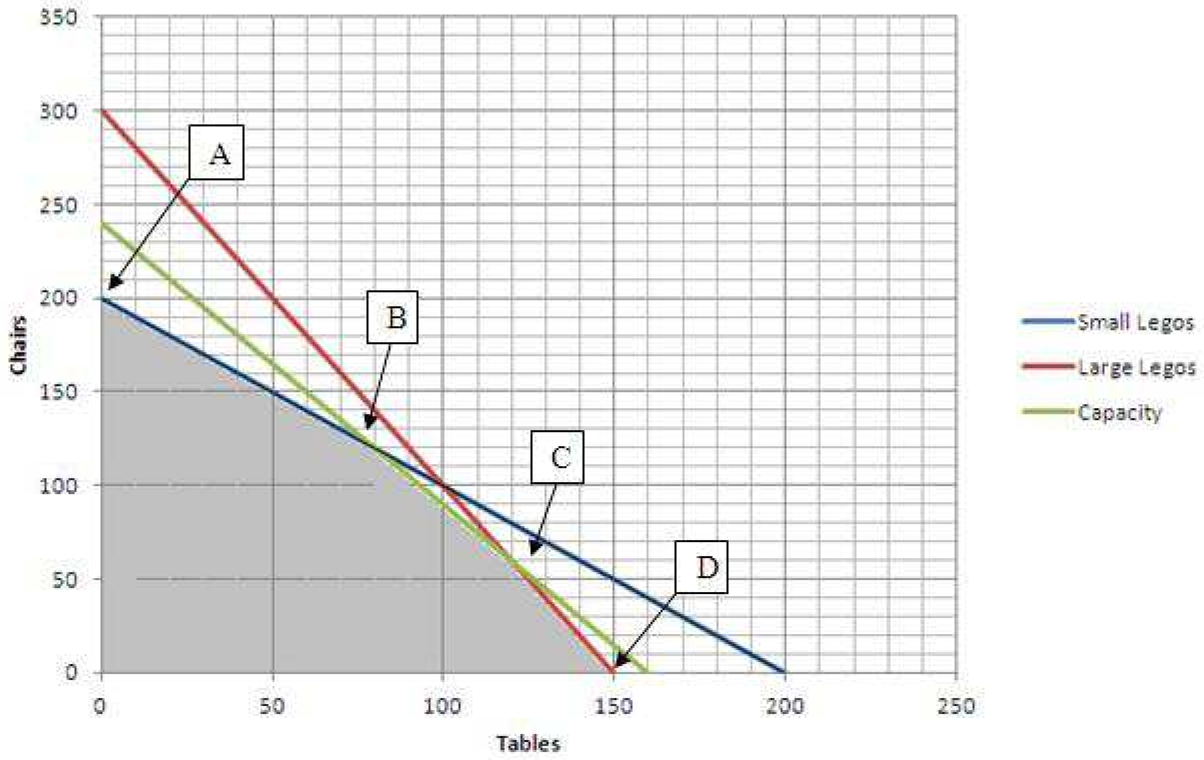


Figure 3.3 shows the addition of the capacity constraint. The constraint is 6 TAB + 4 CHR <=960. When TAB = 0, CHR = 960/4 = 240. When CHR = 0, 6 TAB = 960, TAB = 160. So the line passes through (0,240) and (160,0). It is also less than or equal to, and as previously, all points on the line and back toward the origin are feasible.

Figure 3 also identifies the four EXTREME POINTS (A, B, C, and D) that define our final feasible region after all three constraints have been graphed.

3.3.2 Finding and Evaluating the EXTREME POINTS

Earlier, it was mentioned that extreme points are the candidates for the optimal solution. In this example, our extreme points represent specific

levels of table and chair production. But why just consider the extreme points? What about other possible solution inside in the feasible region?

Many text books will give you a long-winded (and certainly accurate) presentation about something called the “iso-profit” approach in solving LP models and all kinds of other explanations about the process of this final step at determining the optimal solution. There is nothing wrong with their presentations. My preference is to ask you to trust me, get comfortable with our straight-forward way of finding the best extreme point (my opinion), and move on quickly to solving bigger problems using EXCEL.

What do I want you to trust me about? The optimal solution of the model will lie at one of these extreme points (ignoring the possibility of *multiple optimal solutions*, which we will discuss in the next Module). This is due to having a linear objective function (with linear constraints). If we were to plot different levels of objective function values, the line representing the optimal level would be tangent to the feasible region at one of the extreme points (the optimal point!). Use of the word *tangent* may make you think of differential calculus, and there is some similarity to what is occurring here in terms of finding the optimal point. The punch line: All we need to know is the decision variables' values at the extreme points, and then we can use that to find the optimal solution of our LP model. If that isn't a good enough explanation, or it doesn't sound intuitive, contact me and we can talk a little more.

So we will now identify the table and chair production levels of these extreme points. Points A and D, as axis intersection points, have already been determined. Point A is the intersection of the CHR axis (where TAB=0) and the small LEGO constraint. We earlier found this point to be (0,200) – 0 TAB and 200 CHR. Similarly, Point D is the intersection of the TAB axis (CHR = 0) and the large LEGO constraint. We earlier found this point to be (150,0). Two points found, two more to go.

Point B in our graph is found at the intersection of the small LEGO constraint and the capacity constraint. Point C is the intersection of the large LEGO constraint and the capacity constraint. A side note: The intersection point between the large LEGO constraint and the small LEGO constraint falls outside the feasible region. Thus, it is not a possible extreme point that we need to worry about because it is not feasible. In case you were wondering!

So, back to the main broadcast: When we have two lines intersecting – the point of intersection is defined by treating the intersecting lines as a system of linear equations. When we solve for the common variable values, we have found the intersection point, and thus, the corresponding number of tables and chairs that it represents. So, for Point B – the two lines that intersect:

2TAB + 2 CHR = 400

6TAB + 4 CHR = 960

Rewrite the capacity constraint as 3 TAB + 2 CHR = 480 (we simply divided the entire equation by 2). Rewrite it again as 2 CHR = 480 – 3 TAB. Substitute (480 – 3 TAB) in the first equation for the value of 2 CHR. This gives us the first equation rewritten as:

2TAB + (480 – 3 TAB) = 400.

Get variables on one side, numbers on the other, jokers to the right, we get:

–TAB = –80 or TAB = 80.

We can then back substitute TAB = 80 into any form of the previous equations – using 2TAB + 2 CHR = 400, it simplifies to 2(80) + 2CHR = 400. 2 CHR = 240, which implies CHR = 120.

So Point B represents producing 80 tables and 120 chairs (80,120).

Point C is found in similar fashion.

The lines of the two constraints that intersect at Point C:

2TAB + CHR = 300

6 TAB + 4 CHR = 960.

From the first equation, CHR = 300 – 2 TAB.

Replace CHR in second equation with (300 – 2TAB) to get 6 TAB + 4(300 – 2TAB) = 960.

This gives us 6TAB + 1200 – 8TAB = 960.

Numbers to the left, variables to the right, stuck in the text with me you get:

240 = 2 TAB or TAB = 120.

Back substituting we then solve for CHR: CHR = 300 – 2(120) = 60.

So, Point C represents producing 120 tables and 60 chairs.

Did my double Stealers Wheel references clear the copy editor?

After all of this, we now have four candidate production amounts that could be the best solution – the table below compares them. How do we determine the best? We simply take the production levels and using the objective function values (the per-unit sales data), find the total sales amounts. And the best one wins!!

The table below calculates the total sales. Recall that each table contributes $16 to sales and each chair $9.

Extreme Point	**Tables – Sales = $16**	**Chairs – Sales = $9**	**Total**
A	0	200	1800
B	80	120	2360
C	120	60	2460
D	150	0	2400

So, the optimal extreme point was found to be point C – furniture production of 120 tables and 60 chairs. Thus, this is the optimal solution to the LP model created to analyze this scenario.

Reading Material: 3.4 – Summary/Conclusions

A few comments on our first LP model created and solved:

1. There is no correlation of the distance between an extreme point and the graph origin (0,0) and its likelihood of being the optimal solution point. The objective function coefficients determine which extreme point is optimal. In the next module, we will explore how small (or not so small) changes in sales price, and nothing else, might alter the optimal mix of furniture. For instance, if the sales price of tables was just a little more, we may have determined that it is best to just produce 150 tables (Point D). Or, if the sales price of tables was just a little lower, we may have found the optimal production mix to be 80 tables and 120 chairs (point B). The constraints define the possible extreme points, but the objective function identifies the best solution. Subsequent modules will further clarify these types of insight-related discussions through what is traditionally referred to as Sensitivity Analysis.
2. There are a lot of unique ways in which one can execute the mechanistic steps involved in solving a LP model by paper and pencil in two dimensions. If you have your own process (and it follows the laws of algebra) – go for it.
3. Another reason I'm not too concerned about elegance in finding graphical solutions – while we want to build good, sound foundational skills in modeling, we really want to do it expediently so we can advance quickly to more realistic-sized applications of LP. Real problems are quite a bit more complex than those with only two decision variables – which is the only scenario in which solutions can be found graphically. This is still very important – just not to linger too long.

I think it is worth noting that when I talk to those who have successfully completed our class, many reflect on the time we spend building this strong foundation as a key to their long-term success in understanding modeling, interpreting, and gaining insight. They find this to be true as the course moves to larger, more realistic-sized models and when faced with scenarios amenable to such approaches in practice.

A few foundational practice problems follow. Some involve setting up the LP model first, then solving. A few of the problems give you a mathematical model and ask you to solve it – reinforcing just the mechanics of the solution process.

Then, in the next module, we analyze other aspects of the optimal production mix we just found to illustrate some additional problem insight that can gleaned from the model. We previously mentioned that it may be helpful to understand what impact small (or big) changes in sales prices might have on our optimal production mix (this could be due to error, change in market conditions, or anything). Other relevant questions might be: What if our supply of large LEGO blocks was reduced as a result of a union work stoppage? Or, what if we could increase our supply? Or, how valuable are additional units of storage?

These questions dig deeper into the model under analysis beyond "what is the right answer" or the optimal solution. It represents our starting point with assisting decision makers in gaining insight about the scenario under study. It is very important that our decision makers gain this from creating and solving our algebraic LP models. The next module introduces us to some of the insightful issues before we move on to expand our modeling repertoire.