Linear Programming Primer:
Geodesic Domes: Their Flats & Their Corners

(Draft 08-19**)**

The purpose of this note is to introduce you to the world of Linear Programming (LP) by using simple pictures and small numbers! But seriously, this discussion is to introduce you to the pre-eminent mathematical tool of the 20th century, called Linear Programming. Whether by luck or genius, the discovery of this technique has resulted in hundreds of millions (billions?) of dollar savings when trying to allocate resources under constraint.

The first example below is in terms of a 'best' production mix, but the ideas are extremely general. The underlying plan is to quickly find that particular combination of allocation of resources/activities that will maximize some *benefit* or minimize some *cost*. For big problems this results in a giant search problem that Linear Programming cuts down to a very small number of potential best solutions that can easily be examined.

In this note I show how you might set up some simple LP problems, deferring their solutions to a computer package. In the "Cut to the Chase" section though, I do show how you could set up and then visually solve a 2-dimensional LP problem. Beyond 2 or 3-dimensions the solutions become impossible to solve graphically but, the insight gained from the simple 2-D solution carries through to all LP problems of whatever dimension.

Cut to the Chase - A Product Mix Example

Look at the next diagram as an example of a 2-variable Linear Programming problem. I'll go into lots more detail a little later, but first, just get a feel for the question and what would constitute an answer. The situation is that I manufacture a certain number of chairs and tables (the two variables) and I calculate that I have a little excess capacity available that I *could* use to make a few more chairs and/or tables. (Sales tell me I can sell everything I can make, but then, that's sales)! The question is: how many extra chairs and tables *should* I make so as to maximize my profit margin, subject to my production capacity constraints? (Check out the Operations Research text by Hillier and Lieberman for an in-depth discussion and explanation of Linear Programming)

Suppose I know that my profit margin is $3 for every chair I make and $5 for every table. Putting this in equation form will result in that is called the *objective* function. In this case, my objective is to maximize that function by making the best mix of chairs and tables, subject to constraints.

The three relevant production areas that have capacity constraints are: my chair group can make no more than 4 (extra) chairs per hour and my table group not more than 6 (extra) tables. Both the chair and table group must route their production through the assembly group. The assembly group's capacity is expressed by saying that for every chair they assemble it takes 3 units of capacity, while an assembled table takes 5 units. Assembly has a total of 18 units of capacity.

Ok check out the diagram below. For now just gaze at the various lines displayed. The thin lines express the constraints (boundary conditions) while the bold line expresses the objective function. Notice that the axes form part of the boundary as well. The dappled area is called the *feasible region* since *every allowable combination* of chairs and tables must fall within those boundary lines. Solutions like 0 chairs and 6 tables are in here as well as 2 chairs and 6 tables. The function I want

to maximize, Z= 3 * x1 + 5 * x2 can be drawn as a line which can be *moved parallel* to its direction until it just touches the farthermost corner of the feasible region. Notice I have shown four positions of the object function line. The plan is to move it to the best corner.

- Z = 4 * $3 + 0 * $5 = $12 (make only 4 chars, no tables)
- Z= 4 * $3 + 3 * $5 = $27 (make 4 chairs and 3 tables, notice that this maxes the assembly capacity)
- Z= 0 * $3 + 6 * $5 = $30
- Z = 2 * $3 + 6 * $5 = $36*** (optimal solution, 2 chairs and 6 tables)

That's the LP solution and occurs at the *corner point* that represents: 2 chairs and 6 tables. No other allowable combination will do better than this $36.

That's it - the best solution is always found at a corner point (or along a line joining two adjacent corner points).

In three dimensions (three variables) the constraints and their intersections construct a *geodesic dome* like structure with its corners indicating all the best possible solutions and the flat plates represent the constraints, represented by planes. Notice that for a geodesic dome, those flat plates intersect to form lines and points, same for the LP solution space.

In higher dimensions the intersection of the constraints construct what is called a (convex) polyhedron which has lots more corners but the solution approach is the same: the corners (or the line joining them) are where you will find the points that maximize or minimize the objective function.

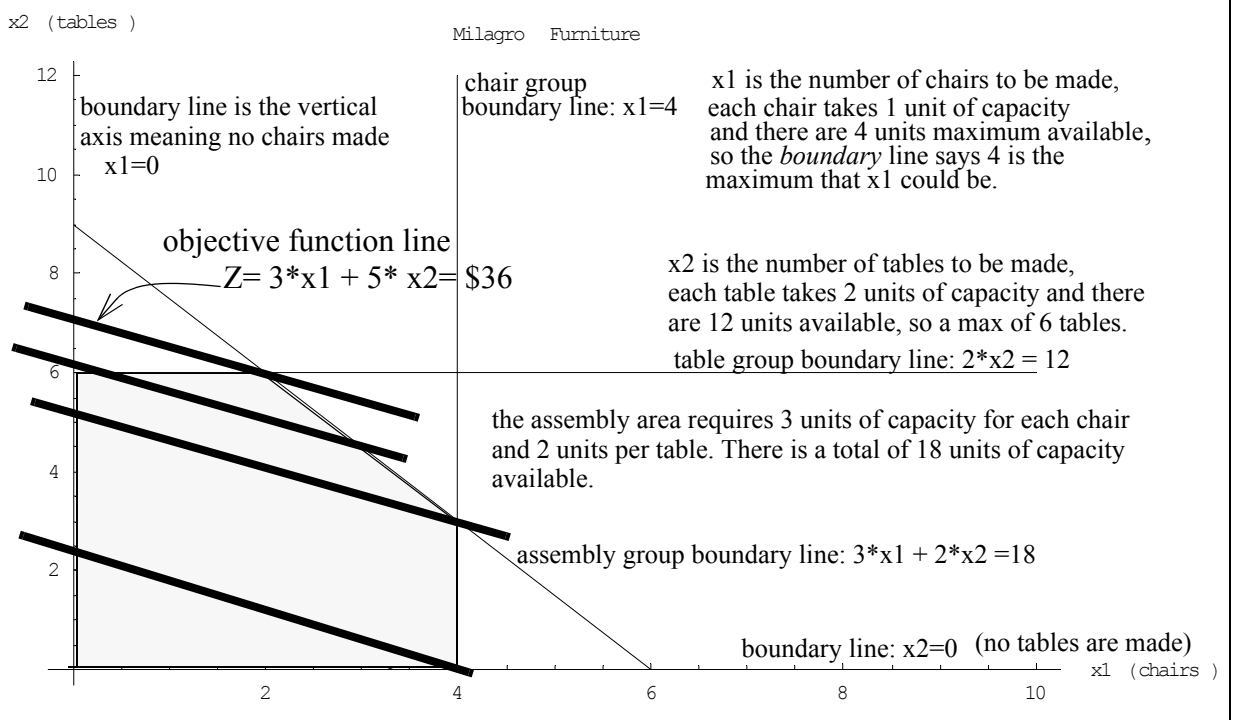

FIGURE 1. Product Mix LP Example - Milagro Furniture

Linear Programming & Transportation

Early uses of LP were, and still are, concerned about getting from here to there in the most economical manner. All forms of transportation from pipeline network protocols to trucking delivery routes have been recast as LP problems. The example below is a (majorly simplified) classic.

Transportation Example

Consider a manufacturer with two factories F1, and F2 that supply two distributors, D1 and D2. The objective is for the factories to supply the distributors at minimum transportation cost. The picture says it all and you could plug in various numbers and eventually arrive at the minimum cost to honor all the demands, given the factory capacities.

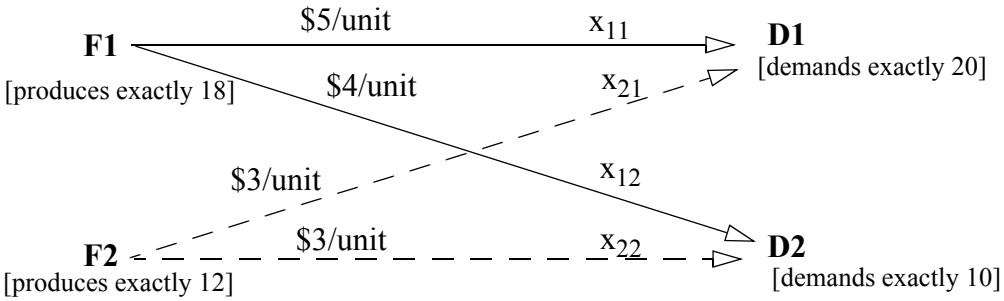

FIGURE 2. Transportation Example: Factory to Distributor Transport Charges

We see that distributor D1 demands 20 units per month from the factory while demand from D2 is 10 units per month, exactly. The factories together can produce the required 30 units per month with factory F1 producing 18 units and F2 producing 12, exactly. The trick is to allocate that factory production to the two distributors in such as way that the shipment costs are minimized while still honoring their demands.

Let me introduce a little notation at this point which is standard in the Linear Programming field.

The symbols $x_{i,j}$ are called *decision variables* and in this case are the number of units to be shipped from location "i" to location "j". For example:

x_{11} is the unknown number of units that I am going to ship from F1 to D1. (Note: x_{11} as well as the other variables could be zero)

x_{12} is the unknown number of units that I am going to ship from F1 to D2.

Note: $x_{11} + x_{12}$ must total to 18 since the output from F1 totals 18 and must go to D1 or D2 or both.

x_{21} is the unknown quantity that I am going to ship from F2 to D1

x_{22} is the unknown quantity that I am going to ship from F2 to D2

Note: $x_{21} + x_{22}$ must total to 12 since that is the output of factory F2.

Note also that: $x_{11} + x_{21}$ must total to 20 in order to honor D1's demand

$x_{12} + x_{22}$ must total to 10 in order to honor D2's demand

As a last requirement, all of these variables, $x_{i,j}$, must be non-negative, that is, the distributors can't ship items back to the factories! These are called the *non-negativity constraints*.

My objective is to minimize cost, while at the same time satisfying these demand and factory ca-

pacity constraints. Let me write this objective in terms of a cost function, called the transportation cost function:

5[$/unit] * x_{11} [units] + 4[$/unit] * x_{12} [units] + 3[$/unit]* x_{21} [units+3 [$/unit] * x_{22} [units]

Let me run through a couple of the terms of this cost function since the units are important and will help you see what is needed.

5 * x_{11}: says that for every unit shipped from F1 to D1, there will be a cost of $5

4 * x_{12}: says that for every unit shipped from F1 to D2, there will be a cost of $4

similarly for the last two terms.

TABLE 1. Transportation Cost Matrix - Cost *from* Factory *to* Distributor

Factory/Distributor	distributor D1	distributor D2	Total Factory Units Produced
F1	$5/UNIT	$4/unit	18
F2	$3/unit	$3/unit	12
Total Demanded Units	20	10	

Stating the Linear Programming Problem

When talking about the math representation of the problem above, it is usually presented as below.

Minimize 5* x_{11} + 4* x_{12} + 3* x_{21} +3 * x_{22}

subject to the following constraints:

$x_{11} + x_{12} = 18$

$x_{21} + x_{22} = 12$

$x_{11} + x_{21} = 20$

$x_{12} + x_{22} = 10$

$x_{11} >= 0$

$x_{12} >= 0$

$x_{21} >= 0$

$x_{22} >= 0$

Exercise For the Above Transportation Example

Given the transportation problem presented above, try assigning various numbers of shipped units so that total transportation costs are minimized. For example, you might decide to ship all of factory 1's (F1) output, 18 units, to D1. This would leave D1 short 2 units that would have to be made up by factory F2. So, F2 must ship 2 units to D1 and 10 units to D2.

This allocation of shipment units would result in a total cost of

$5* 18 + $4* 0 + $3* 2 + $3 * 10 = $126 (can you do better?) Try it!

In practice, these problems are solved by computer packages that can handle hundreds if not thousands of variables and equations. I show how a very simple problem can be solved *visually* in the next section which is extremely helpful in understanding the method but is very limited in actual computation. For the above exercise I used a function within the *Mathematica* programming pack-

age that does all this in milliseconds.

From the package, I get $116 for the minimum total cost, with a solution vector of {8,10,12,0})

That is:

$x_{11} = 8$

$x_{12} = 10$

$x_{21} = 12$

$x_{22} = 0$

You may be getting the idea that trying various combinations would take a lot of your time before finding the lowest cost, not to mention frustration as to whether what you found was actually the lowest possible. Linear programming routines solve giant problems like this thousands of times daily and guarantee a minimum cost solution (assuming all of the constraints and assumptions hold). So, if you are going to contribute to resource allocation tasks within an organization, Linear Programming is an essential skill set to master.

Cut to the Chase Re-Visited

Let me give a little more detail, and a slight revisions, for the "Cut to the Chase" example above. Remember that this was a *product mix* question where I try to *maximize profit* by making two products that are constrained by plant capacity. How can I *allocate resources among activities* so as to maximize economic returns. Once the constraint lines are drawn, which are simply lines, the LP algorithm geometrically moves from corner to corner until the objective function can't be increased further. The corners I am talking about are the intersections of the constraint lines.

Suppose I own a factory, "Milagro Furniture" and have three production areas. Area "C" makes basic lawn chairs, Area "T" makes basic patio tables, while Area "Z" assembles, customizes, and prepares for sale the chairs and tables from the other two areas.That is, all of the production of areas C and T must go through area Z.

I am thinking about introducing two possible new products, a new type of patio chair and a new type of patio table. There is some limited production capacity available in all three areas that I can devote to the new products, although the area Z capacity is linked to that of areas C and T and so is not independent.

Marketing tells me that I can sell all I can make of the new products!

Now, I have to decide how much of each area's capacity to allocate to the new products. Capacity in this case is measured by "capacity used per unit production rate". I have calculated this for each area such that if I produce one chair per hour in the "C" area, I will use 1% of my production capacity of that area. If I produce 1 table per hour in area T, I use 2% of the production capacity in the "T" area. Similarly for the customizing area, customizing 1 chair and 1 table would take 3% per chair and 2% per table.

I have available (or can make available) the following excess capacities:

- up to 4% in area C (chair production)
- up to 12% in area T (table production)
- up to 18% in area Z (the customizing area)

I know I can make a net profit of $3 for each new chair and $5 net for each unit of the new type of

table. For example, if I *could* complete 4 chairs and 6 tables per hour, I would net:

$3 * 4 + $5 * 6 = $42/hour. (Unfortunately we will see that that mix is not possible, given the constraints. That is the challenge that linear programming helps to resolve, what is the *best* I can do, given the constraints I need to work under.)

Notation:

x_1 is the number of the new type of chair planned per hour

x_2 is the number of the new type of table planned per hour

I want to maximize my net profit per hour by planning and building as many new chairs and tables per hour as I can, given my capacity constraints. So I want to maximize the value of "Z" in the equation below.

Maximize Z= $3*$x_1$ + $5*$x_2$

subject to:

eq1: 1*x1 <= 4 (it takes 1 unit of capacity to make 1 chair per hour in area "C" and there are up to 4 units available)

eq2: 2*x_2 <= 12 (it takes 2 units of capacity to produce one table per hour, in area "T", and there are up to 12 units available)

eq3: 3*x_1 + 2*x_2 <=18 (it takes 3 units of capacity per chair and 2 units of capacity per table in the customization area "Z" and there up to 18 units of capacity in area Z.)

Further, I know that the available production rates can be considered non-negative. That is,

eq5: x_1 >= 0

eq6: x_2 >= 0

In the diagram below, the solid lines are boundary constraint lines. In this case, the feasible (obtainable) solutions are bounded by these lines. So, for example, x1 must be less than or equal to 4 and x2 must be less than or equal to 6. In addition, 3* x1 + 2* x2 must be less than or equal to 18. So, the allowable solutions must satisfy all three of these constraints as well as being non-negative. The dashed line is a "test" line associated with the objective function "Z". If you visually move that test line upward you will see that it finally touches just one of the 'corner' points. Namely the corner point at {2,6}. This is the maximum that Z can reach ($36) and is the maximum of the 'objective' function.

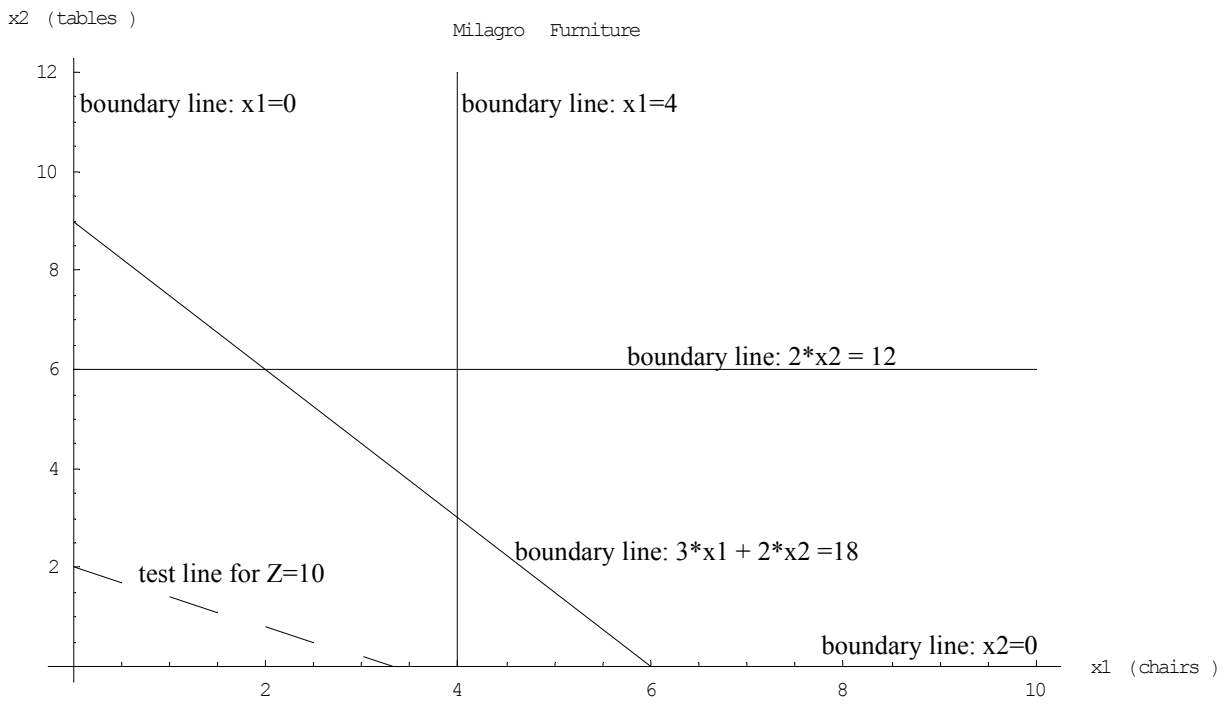

FIGURE 3. Product Mix LP Example - Milagro Furniture

Summary

This note described a couple of extremely simple linear programming formulations. I showed how to visually solve a 2-dimensional problem by drawing lines on a grid. Larger problems require a computer package.

References

Hillier & Lieberman (1985) *Operations Research*