**Sec 8.6: Linear Programming Model**

Linear Programming is a way to solve maximum and minimum problems for businesses (maximize profit or minimize cost). We do this through graphing constraints and then checking the vertices for max or minimum values.

Example 1: Given the following constraints:

Graph each of the constraints and find the vertices. Remember to find where two lines cross use substitution or elimination.

(The x and y axis are also part of the constraints)

Determine the area that meets the constraints.



Determine the vertices of the area. They should be at

You know determine the max and min by using what is called an objective function. For this situation the objective function is . We determine which points are max or min by plugging in each value to the objective function. So . From this we know that at the point (0,4) there is a maximum and at the point (0,0) there is a minimum.

Example 2: You can build a small building using 10 sheets of plywood and 15 studs for a profit of $400 each; a large building can be built using 15 sheets of plywood and 45 studs for a profit of $500 each. At the local hardware store they have on supply 60 sheets of plywood and 135 studs. How many of each kind of buildings should you build to maximize your profit? What would be you maximum profit. Let s represent small buildings and l represent large buildings. We begin by writing out our constraint equations:

Make sure that you understand where the equations come from. Now we graph this information

(I graphed it as s as the x-axis and l as the y-axis)

Determine the area that meets the constraints



Now we find the intersection points (remember to use substitution or elimination). They are at

Now we need our objective function. Remember that we are trying to find out how many buildings we must build to maximize our profit. So our objective function should deal with profit.

After plugging in our vertices we find out that we should build 6 small buildings and no large buildings for a maximum profit of $2400.