Linear Programming (poggers)

Very shortly put, it is simply optimizing a linear function w.r.t. linear constraints.

The matrix form of a maximization problem would be:

$$\text{Maximize } c^T x \tag{1}$$
$$\text{Subject to } Ax \leq b \tag{2}$$

Or, they could be split into many constraints instead, with both $\leq$, $\geq$, and $=$ between them.

The decision variables are x. An assignment to them is a solution. A solution satisfying the constraints is feasible. A feasible solution that actually minimizes/maximizes is optimal (sometimes denoted by an asterisk, i.e. $P*$).

The set of feasible solutions, called F, is a convex polyhedron. That means it has n sides and has no two points such that a line may be drawn between them that crosses outside of the polyhedron. Usually, you need to traverse to a corner of the polyhedron to attain an optimal solution.

Two stinky cases: unbounded solutions, where an arbitrarily good solution may be found, and infeasible LP's, where the set of feasible solutions is empty.

I know what you're thinking at this point: "That's really cool, Jonathan, but what is the standard form of a linear program?" Well, here it is!

$$\text{Maximize } c_1x_1 + c_2x_2 + ... + c_nx_n \tag{3}$$
$$\text{Subject to} \tag{4}$$
$$a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \leq b_1 \tag{5}$$
$$a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \leq b_2 \tag{6}$$
$$... \tag{7}$$
$$a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \leq b_m \tag{8}$$
$$x_1, x_2, ..., x_n \geq 0 \tag{9}$$

Standard form is practical because then you can perform algorithms which assume that your program is of some standard form. Any other linear program can be transformed to standard form with very simple steps. E.g. if we want to minimize, replace with maximizing the negative. If we have some constraint with the inequality flipped the wrong, way, multiply each side with -1, and so on.

The most important one is that you can have an equality and make it into standard form by rewriting it with a $\leq$, and then multiplying each side by -1 WHILST keeping the inequality the same. This way you have two inequalities that, when they are both satisfied, the solution must satisfy the original equality.

A few more conversion tricks:

- If you want an unconstrainted y, replace $y \geq 0$, with $y_+, y_- \geq 0$ and replace y with $y_+ - y_-$ everywhere.
- If you want to adjust a constraint, e.g. to say that $y \geq -17$, replace $y \geq 0$ with $y' = y + 17$. Replace y with $y' - 17$ everywhere.

After you made those substitutions, you can convert a solution found back into a solution for the normal LP. This is done by subbing back $y = y_+ - y_-$. Subbing the other way can be done by subbing $y_+ = \max(y, 0)$ and $y_- = -\min(y, 0)$.

Unfortunately this causes a 2x blowup in size in terms of variables and constraints, but in one of the exercises it's shown that you can do it with just 1 extra constraint & variable.

So what's the simplex algorithm?

The simplex algorithm is basically starting in some corner of F (the polyhedron of feasible points), and then while some better corner exists, go to that. What about interior solutions? Nah moight don't worry about it. I guess that's treated later in the course?

Before going in depth with the simplex algorithm, let's have a lookie at dictionaries.

Dictionaries moight

Aw yiss moight. Dictionaries are a system of equations like

$z =$			$3x_1$	$+$	$2x_2$
$x_3 =$	10	$-$	$3x_1$	$-$	x_2
$x_4 =$	6	$-$	$2x_1$	$-$	$5x_2$
$x_5 =$	2	$+$	x_1	$-$	$2x_2$

The above is constructed from an LP. The LP:

$$
\begin{aligned}
\text{Maximize } & 3x_1 + 2x_2 && (10)\\
\text{Subject to } & && (11)\\
& 3x_1 + x_2 \leq 6 && (12)\\
& 2x_1 + 5x_2 \leq 10 && (13)\\
& -x_1 + 2x_2 \leq 2 && (14)\\
& x_1, x_2 \geq 0 && (15)
\end{aligned}
$$

In the dictionary, there are more variabes than in the original LP. This is because your mom ate them in the original LP (she so fat she has her own gravitational field). The new variables are called slack variables.