Linear Programming Notes III: A Simplex Algorithm Example

These notes will take you through a computation using the simplex algorithm. The example will give you a general idea of how the algorithm works. Except for a few exercises that I give you, you will never do simplex algorithm computations by hand. Why should I bother telling you about it? The basic idea is simple, intuitive, and powerful. That makes it a good idea, one worth knowing something about. The technique is straightforward. In principle, you can learn how to solve any linear programming problem by hand. Algorithms are a central concept in Operations Research. You get an understanding of the general concept by seeing a particular example.

I will start by pushing through a specific example. When I am done, I will make remarks about what must be done to come up with a general algorithm.

$$\begin{array}{lrcrcrcrcrl}
\text{max} & 2x_1 & + & 4x_2 & + & 3x_3 & + & x_4 & & & (0) \\
\text{subject to} & 3x_1 & + & x_2 & + & x_3 & + & 4x_4 & \le & 12 & (1) \\
& x_1 & - & 3x_2 & + & 2x_3 & + & 3x_4 & \le & 7 & (2) \\
& 2x_1 & + & x_2 & + & 3x_3 & - & x_4 & \le & 10 & (3) \\
& & & & & & & x & \ge & 0 &
\end{array}$$

The problem is feasible (since all constraints hold if you set each variable equal to zero). The problem is bounded (inequality (1) implies that $x_1 \le 4$, $x_2 \le 12$, $x_3 \le 12$, and $x_4 \le 3$, for example); hence it is impossible to make the objective function larger than $2(4) + 4(12) + 3(12) + 1(3) = 95$. Of course, we cannot set $x_1 = 4$, $x_2 = 12$, $x_3 = 12$, and $x_4 = 3$ and still satisfy constraint (1), but 95 is certainly an upper bound to the value of x_0.

These observations indicate that the problem has a solution. Finding it does not look easy. In particular, we have too many variables to graph.

Here is the essential approach. First introduce slack variables and write the problem as a system of equations involving nonnegative variables.

$$\max x_0$$

subject to:

$$\begin{array}{ccrcrcrcrcccccccrl}
x_0 & - & 2x_1 & - & 4x_2 & - & 3x_3 & - & x_4 & & & & & & & = & 0 & (0) \\
& & 3x_1 & + & x_2 & + & x_3 & + & 4x_4 & + & x_5 & & & & & = & 12 & (1) \\
& & x_1 & - & 3x_2 & + & 2x_3 & + & 3x_4 & & & + & x_6 & & & = & 7 & (2) \\
& & 2x_1 & + & x_2 & + & 3x_3 & - & x_4 & & & & & + & x_7 & = & 10 & (3) \\
& & & & & & & & & & & & & & x & \ge & 0 &
\end{array}$$

This system of equations is identical to the original system of inequalities. The simplex algorithm is a systematic way of solving the system of equation in a way that:

1. Preserves non-negativity of the variables.

2. Assigns positive values to only a few (one for each equation) variables (the basis).

3. Identifies whether the value of x_0 is as large as possible and if it is not

4. Describes how to increase the value of x_0.

Looking at the system above, it becomes easy to "guess" values of $x_0, \ldots, x_7$ that satisfy the equations. Simply set the variables that appear in more than one equation equal to zero ($x_1 = x_2 = x_3 = x_4 = 0$) and solve for the remaining variables ($x_0 = 0, x_5 = 12, x_6 = 7, x_7 = 10$). This "guess" satisfies the constraints of the original problem. We can increase x_0 however. Look at (0). This equation gives x_0 in terms of $x_1, x_2, x_3,$ and x_4. Because the coefficients of x_1 through x_4 are negative, increasing the value of any one of these variables from 0 to a positive number will increase x_0. It follows that the guess in which $x_1 = x_2 = x_3 = x_4 = 0$ is not a good way to maximize x_0. Fix one of these variables, say x_1. How far can we increase x_1 without violating the constraints of the problem? That is, how far can we increase x_1 without violating constraints and while still maintaining $x_2 = x_3 = x_4 = 0$? Since $x_5 \geq 0$, (1) says that $x_1 \leq 4$; since $x_6 \geq 0$, (2) says that $x_1 \leq 7$; since $x_7 \geq 0$, (3) says that $x_1 \leq 5$. Hence, I can make x_1 as large as 4, but no larger (without changing the values of the variables $x_2 = x_3 = x_4 = 0$ or violating one of the constraints of the problem. The next step of the procedure involves using (1) (the equation that bounds the value of x_1) to eliminate x_1 from the other equations. That bit of algebraic manipulation will yield a system of equations equivalent to the original system (and hence the original problem) that can be solved easily for x_1. First, fix equation (1) so that the coefficient of x_1 is 1. You can do this by dividing the equation by 3 to get equation

$$(1)' : x_1 + \frac{1}{3}x_2 + \frac{1}{3}x_3 + \frac{4}{3}x_4 + \frac{1}{3}x_5 = 4.$$

Next use $(1)'$ to rewrite the other equations in terms of variables other than x_1. That is, write

$$(0)' = (0) + 2(1)';$$

$$(2)' = (2) + (-1)(1)';$$

$$(3)' = (3) + (-2)(1)'.$$

This transformation yields the second representation of the original problem as a system of equations:

Row	*Basis*	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	*Value*
$(0)'$	x_0	1	0	$-\frac{10}{3}$	$-\frac{7}{3}$	$\frac{5}{3}$	$\frac{2}{3}$	0	0	8
$(1)'$	x_1	0	1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{4}{3}$	$\frac{1}{3}$	0	0	4
$(2)'$	x_6	0	0	$-\frac{10}{3}$	$\frac{5}{3}$	$\frac{5}{3}$	$-\frac{1}{3}$	1	0	3
$(3)'$	x_7	0	0	$\frac{1}{3}$	$\frac{7}{3}$	$-\frac{11}{3}$	$-\frac{2}{3}$	0	1	2

I have introduced a shorthand notation for systems of equations. Rather than repeat the variables that appear in each equation, I wrote them along the top. You should interpret each line $(0)'$ through $(3)'$ as an equation (the equation $(1)'$ is expanded above). The first column in the table indicates the row number. The second column indicates the basis. Notice that the basis variable appears in its equation with coefficient one and in no other equation (or with coefficient zero in the other equations for you purists). The value column tells you the right hand sides of each of the equations. This second system of equations can be solved easily for x_0, x_1, x_6, and x_7 if we set the remaining variables equal to zero. We obtain (by setting basis variables equal to corresponding values): $x_0 = 8, x_1 = 4, x_6 = 3, x_7 = 2$. In terms of the original problem all that we have done is said that we can increase x_0 from 0 to 8 if we set $x_1 = 4, x_2 = x_3 = x_4 = 0$. The good news is that we made this observation in a systematic way. We can repeat the process. As x_2 appears in $(0)'$ with a negative coefficient, increasing x_2 (from 0) increases x_0. How great can this increase be (if we insist that the variables that appear in several equations - the non basic variables - be set equal to zero)? $(1)'$ allows $x_2 = 12$ (since we need $\frac{1}{3}x_2 \leq 4$); $(2)'$ places no restriction on x_2 as x_2's coefficient in that constraint is negative. If we increase x_2 we can also increase x_1 to "take up the slack;" $(3)'$ requires that $\frac{1}{3}x_2 \leq 2$ or that $x_2 \leq 6$. Since the $(3)'$ constraint places the strictest restriction on x_2, we use $(3)'$ to eliminate x_2 from the other equations and come up with a solution to the equations in which x_0 goes up (from 8) and x_2 is set equal to 6. The transformed equations satisfy:

$$(3)'' = (3)(3)'$$

and then

$$(0)'' = (0)' + \frac{10}{3}(3)'';$$

$$(1)'' = (1)' - \frac{1}{3}(3)'';$$

$$(2)'' = (2)' + \frac{10}{3}(3)''.$$

The following array summarizes the computation.

Row	*Basis*	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	*Value*
$(0)''$	x_0	1	0	0	21	−35	−6	0	10	28
$(1)''$	x_1	0	1	0	−2	5	1	0	−1	2
$(2)''$	x_6	0	0	0	25	−35	−7	1	10	23
$(3)''$	x_2	0	0	1	7	−11	−2	0	3	6

Here we can take $x_3 = x_4 = x_5 = x_7 = 0$ and solve for the rest of the variables:

$$x_0 = 28, x_1 = 2, x_6 = 23, x_2 = 6.$$

Again we have an equivalent representation of the original system. Again we have increased x_0 (this time to 28). Again we can see that it is possible to increase x_0 further by increasing x_4 (at this point increasing x_3 would actually lower the objective function value). Arguing as before, it is $(1)''$ that restricts the increase of x_4, while the other equations do not restrict x_4. Hence, I use $(1)''$ to eliminate x_4 and form a new system: $(1)''' = \frac{1}{5}(1)''; (0)''' = (0)'' + (35)(1)'''; (2)''' = (2)'' + (35)(1)'''$; and $(3)''' = (3)'' + (11)(1)'''$. This new system looks like:

Row	$Basis$	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$Value$
$(0)'''$	x_0	1	7	0	7	0	1	0	3	42
$(1)'''$	x_4	0	.2	0	$-.4$	1	.2	0	$-.2$	.4
$(2)'''$	x_6	0	7	0	11	0	0	1	3	37
$(3)'''$	x_2	0	2.2	1	2.6	0	.2	0	.8	10.4

Now observe that as before we can read off a feasible guess to the original problem: $x_0 = 42; x_1 = 0; x_2 = 10.4; x_3 = 0; x_4 = .4; x_5 = 0; x_6 = 37$; and $x_7 = 0$. This time we have actually solved the original problem. Here is why. Equation $(0)'''$ expresses x_0 in terms of x_1, x_3, x_5, and x_7. Further, the coefficients of these variables are all non-negative in $(0)'''$. Since these variables must take on non-negative values, the best value that they can take on (for the purpose of maximizing x_0) is zero. If you set $x_1 = x_3 = x_5 = x_7 = 0$, then you get 42. More mathematically, rewriting $(0)'''$ yields

$$x_0 = 42 - (7x_1 + 7x_3 + x_5 + 3x_7).$$

The stuff in parentheses is always nonnegative since $x_1, x_3, x_5, x_7 \geq 0$. It follows that $x_0 = 42-$ blah, where blah is a non-negative number. Therefore, $x_0 \leq 42$ in any possible solution involving a feasible vector for the original problem. Since I have found a way to make $x_0 = 42$, I must have found a solution to the original problem. That solution is: $x_1 = 0, x_2 = 10.4, x_3 = 0$, and $x_4 = .4$.

This example explains the essentials of the simplex algorithm. Here are some comments. First, although I began by increasing x_1's value, I ended with $x_1 = 0$. Thus, what looked like a good idea at first did not turn out to be a good idea. The procedure that I have described always (step by step) increases x_0, but there is not much you can say about how it changes the values of other variables. Second, I always increase the value of some variable that appears with a negative coefficient in row 0. The variable that I select will have no influence on the final outcome. I could have started by increasing x_2, x_3, or x_4 instead of x_1. In fact, if I had started with either x_2 or x_4 it would have been possible to obtain the solution in two steps instead of three. Third, I decided which row to use in a "pivot" in a systematic way. The values (right-hand side constants) are positive at every step of the computation. This is no accident. Indeed, this property is what guarantees that your "guess" is feasible at each step of the algorithm. Here is a general outline of how the simplex algorithm works. It takes a linear programming problem and, in a finite number of steps (bounded by the number of constraints and variables in the original problem), it stops. When it stops, the algorithm either proves that the problem is infeasible; provides a solution to the problem; or demonstrates that the problem is unbounded.

You start with a description of the feasible region using variables that are constrained to be non-negative. It turns out that it is always possible to do this. That is, given any LP, you can transform it to one in which variables are non-negative and constraints are equations. In the example, we transformed inequalities to equations by adding slack variables. Second, you find a feasible basis for the system of equations. A basis is a subset of variables, one for each equation. You try to find a guess for the problem by solving the constraints using only these variables (setting the other variables equal to zero). In the example, the initial feasible basis consisted of the slack variables and x_0. If you describe your feasible set using n equations, then you should only need n of your variables to solve the equations. This amounts to finding a corner of the feasible set. You may not be able to find a feasible basis for the problem. (After all, the problem may be infeasible.) It turns out that if the problem is feasible, then there exists a feasible basis. The standard way to find a feasible basis is to solve an "artificial" LP derived from your problem.

Here is what you should know about finding feasible bases. Not all LPs have feasible bases. You can find a feasible basis (if one exists) using the simplex algorithm. It is easy to find feasible bases for some problems. For example, in the example, the slack variables were a feasible basis. In general, if the feasible set can be written $Ax \leq b$ and $x \geq 0$ for $b \geq 0$, then $x = 0$ always satisfies the constraints and a good starting feasible basis is the slack variables.

Once you have written the problem in basic form (as in any of the arrays above), you conduct a pivot operation. You do this by looking for a negative number in Row 0. (If there are none, then you have solved the problem.) If there is more than one negative number, pick one. Let us say that it is in column j. You can increase the value of the objective function by "pivoting" x_j into the basis. You must decide where to pivot. Look in the x_j column. If there are no positive numbers in the column, then stop. The problem is unbounded (you can increase x_j indefinitely without violating the constraints of the problem). If there is at least one positive number in the column, then pick the one that minimizes the ratio between the value column and the entry in the x_j column. This is precisely what I did in the computation above. Doing this identifies a pivot element. Now you do a computation that generates a new basis. If the pivot is done correctly, then, when you are done, you still have a basis; the value of the objective function goes up; and the value column remains non-negative. Now you can repeat the pivoting process. The process must end in a finite number of steps because there are only a finite number of bases and you never return to a basis once you have pivoted away from it (because at each step you increase the value of the objective function). This is essentially the entire story of the theory of the simplex algorithm. The story had a few loose ends, but only a few. For the sake of truth, here are the loose ends.

1. Finding an initial feasible basis.

   It requires a bit of cleverness to realize that you can do this (when it is possible) by solving another linear programming problem.

2. Proving that you have an unbounded problem when you have found a negative number in row zero with no positive number in the corresponding column.

   This only requires thinking about what the simplex array means.

3. Proving that if you pivot correctly you obtain a new feasible basis.

   You need to describe the simplex pivoting rules precisely and verify that they do what they are supposed to do. The precise rules are exact translations of what I have said in words. Verifying that everything works is easy.

4. Proving that the objective function goes up after every pivot (thereby guaranteeing that the algorithm stops in a finite number of steps).

   It turns out that the objective function need not go up every step. There are "degenerate" cases (when there is a tie in the minimum ratio rule that determines the pivot row). In these cases, a basis variable may take on a zero value and subsequent pivots may lead to a new basis that leaves the value of x_0 unchanged. These ties are rare. In practice, the implementation of the algorithm breaks ties randomly. In theory, there is a refined method of picking where to pivot. If you follow the refined rule, you are guaranteed to never repeat a basis even in degenerate problems.

5. Finding all solutions.

   The algorithm finds one basic solution to an LP (provided that a solution exists). When there are many solutions, you can use the algorithm to find all basic solutions. This is not hard, but I won't say more about it.