Overview of the simplex method

Steve Hurder

1 The basic problem

This handout describes in a brief form the simplex method from Chapter 8, sections 1 & 2 of the text LINEAR ALGEBRA AND ITS APPLICATIONS, Third Edition, by Gilbert Strang.

We are given variables $x_1, x_2, \ldots, x_n$ which label something that we can measure: output of a factory for example, or material required to make something. It is natural in the problems given to require that the vairables $\{x_i\}$ are never negative. We write this as $x_i \geq 0$.

A *cost function* is given which measures the quantity that we want to be either maximized or minimized. This is written as

$$C(\vec{x}) = C(x_1, \ldots, x_n) = c_1 \cdot x_1 + \cdots c_n + \cdot x_n$$

Finally, there must be constraints on the allowed values of the variables $\{x_i\}$ which reflect the problem. (Without constraints, the max/min problem for $C(\vec{x})$ usually has only stupid solutions like all $x_i = 0$, or no solution at all.) The constraints take the form of equations that require some combination of the variables to be greater than some other constants. Assume there are m constraint equations, which we write as:

$$\begin{array}{cccccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & \geq & b_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & \geq & b_2 \\
\vdots & & \vdots & & \vdots & & \vdots & \vdots & \vdots \\
a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & \geq & b_m
\end{array}$$

The problem is to find the maximum or minimum value of the function $C(\vec{x})$ on the set of points $\vec{x}$ which satisfy the m constraint equations and are also non-negative. This set of points is called the *feasibility set.*

THEOREM 1.1 *When the linear function $C(\vec{x}) = c_1 \cdot x_1 + \cdots + c_n \cdot x_n$ has a maximum (or a minimum) on the fesibility set, then the maximum (respectively, the minimum) must occur at a vertex of the feasibility set.*

Proof. We won't actually prove this, but will describe the proof. The cost function $C(\vec{x})$ is linear, so if we fix the cost at some constant, $C(\vec{x}) = \mathbf{K}$, then this describes a hyperplane in the space of variables $\vec{x} = (x_1, \ldots, x_n)$. (For two variables, a hyperplane is a line in the plane; for three variables, a hyperplane is a plane in three-space; and so forth.) On the other hand, the feasibility set is the intersection of *half-spaces.* These half-spaces are the set of points satisfying just one of the inequalities. So the feasibility set is a simplex in the n-space of the variables. To find the maximum cost, we assume there is some constant $\mathbf{K}$ so that the hyperplane $C(\vec{x}) = \mathbf{K}$ misses the feasibility set completely. (If such a constant does not exist, then there is no maximum.) The maximum cost is then less than this constant $\mathbf{K}$.

So now assume that the hyperplane $C(\vec{x}) = \mathbf{K}$ misses the feasibility set completely. We then decrease the cost slowly by decreasing the value of the constant $\mathbf{K}$. Eventually, the plane $C(\vec{x}) = \mathbf{K}$ must intersect the feasibility set for some value of the constant $\mathbf{K}$. (Otherwise, the feasibility set is empty!) When the hyperplane first touches the fesibility set, we must be on either a vertex, or an edge, or some face of the feasibility set. In any case, we find a vertex for the feasibility set which first intersects the decreasing cost function's hyperplane.

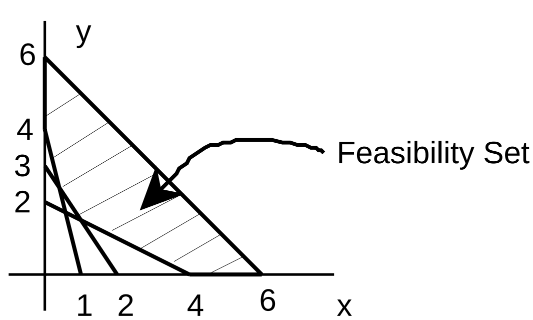
The same type of argument shows that when a minimum exists, it also occurs at a vertex in the feasibility set. □

One way to find the maximum or mnimum is to graph the feasibility set by plotting all of the planes given by the equations above. We then test the cost function at the vertices we find, and take the maximum or minimum values. This works if we are in the plane or in three-space. It is not so easy to do when we have to "plot" in more than three dimensions! Here is an example:

EXAMPLE 1.2 *Find the minimum of the cost function* $C(x, y) = 3x + 2y$ *on the set of points which satisfy the inequalities*

$$x \geq 0; \quad y \geq 0; \quad x + 2y \geq 4; \quad 3x + 2y \geq 6; \quad 4x + y \geq 4; \quad -x - y \geq -6$$

The feasibility set is the region bounded by the lines $x + 2y = 4, 3x + 2y = 6, 4x + y = 4, x + y = 6$ in the first quadrant:



The values of the cost function on the vertices is given by

$$C(0, 6) = 12; C(0, 4) = 8; C(.4, 2.4) = 6; C(1, 1.5) = 6; C(4, 0) = 12; C(6, 0) = 18$$

so that the *minimum* occurs at both (.4,2.4) and at (1,1.5), so all points along this segment of the boundary are minimums with cost 6. The maximum is at (6,0), where the cost is 18.

2 Set-up for simplex method

Given a problem as in section 1 with constraint equations, the simplex method reduces the finding of the maximum or minimum of the cost function to a problem solved by matrix methods. The steps to setting up for the simplex method are:

1. Introduce new variables

$$w_j = x_{n+j} = a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n - b_j \geq 0$$

   for $j = 1, \ldots, m$

2. Arrange the array of numbers $\{a_{ij}\}$ in the constraint equations into a matrix of size $m \times (n+m)$ by adding onto the given array a copy of -I:

$$A = \begin{bmatrix} a_{11} & \ldots & a_{1n} & -1 & 0 & \ldots & 0 \\ a_{21} & \ldots & a_{2n} & 0 & -1 & \ldots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & \ldots & a_{mn} & 0 & 0 & \ldots & -1 \end{bmatrix}$$

   The constraint equations are now written as $A \cdot \vec{x} \geq \vec{b}$, where we have extended $\vec{x}$ to a column vector of height $n+m$: $\vec{x} = [x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}]^T$.

3. Form the matrix tableau, by adding into the "A matrix" the coeficients of the cost function $C(\vec{x})$ and the range values $\vec{b}$:

$$\begin{bmatrix} a_{11} & \ldots & a_{1n} & -1 & 0 & \ldots & 0 & b_1 \\ a_{21} & \ldots & a_{2n} & 0 & -1 & \ldots & 0 & b_2 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \\ a_{m1} & \ldots & a_{mn} & 0 & 0 & \ldots & -1 & b_m \\ c_1 & \ldots & c_n & 0 & 0 & \ldots & 0 & 0 \end{bmatrix}$$

There is a more general type of simplex problem, which allows for the cost function to depend also on the constarints $w_j \geq 0$. This is accomplished by allowing the entries of the bottom row to be non-zero underneath the matrix -I in the tableau. A correctly set-up "simplex method problem" then consists of:

1. a set of variables $x_1, x_2, \ldots, x_{n+m}$
2. an $m \times (n+m)$-matrix A of constraints
3. a column vector $\vec{b} = [b_1, \ldots, b_m]^T$ of range values
4. a cost function $C(\vec{x}) = \vec{c} \cdot \vec{x} = c_1 \cdot x_1 + \cdots + c_{n+m} \cdot x_{n+m}$
5. A matrix tableau

$$\begin{bmatrix} a_{11} & \dots & a_{1n} & -1 & 0 & \dots & 0 & b_1 \\ a_{21} & \dots & a_{2n} & 0 & -1 & \dots & 0 & b_2 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \\ a_{m1} & \dots & a_{mn} & 0 & 0 & \dots & -1 & b_m \\ c_1 & \dots & c_n & c_{n+1} & c_{n+2} & \dots & c_{n+m} & ?? \end{bmatrix}$$

We will see in the next section how to calculate the value of ?? appearing in the lower right-hand corner of the matrix tableau of the generalized problem above.

3 Choosing the free variables

The tableau formulation of the problem, as described in the last section, assumes that we are starting at the vertex $x_1 = 0, x_2 = 0, \dots, x_n = 0$. The reason for this, is that we have the cost function starting at value "0" in the lower right-hand corner of the tableau. This corresponds to the values $x_i = 0$, and consequently reading the tableau equation across the line yields $-w_j = b_j$. The values of the w_j do not enter the standard cost function, so we get a zero for the initial cost at $x_i = 0$.

When we work with the general problem, the cost function does include the variables $w_j = x_{n+j}$, so to find the value of ?? in the lower corner, we must substitute into the cost function the values $x_i = 0$ and $w_j = -b_j$. This gives the bottom corner value

$$?? = -c_{n+1}b_1 - c_{n+2}b_2 - \dots - c_{n+m}b_m$$

We have expanded our list of variables from "n" to "n+m" to incorporate the constraints. The key idea of the simplex method is that to find a vertex, we must group the variables into two camps: the *basic variables* will be set equal to zero. Notice that we start the problem with the original variables $x_1, \dots, x_n$ as the basic variables. The *free variables* are the variables whose value is left open, and there are m of these. What this means is that after we set the basic variables equal to zero, then we are left with m equations the in m-free variables and we can solve these equations for the values of the free variables.

There is a simple reason why the grouping of the variables into "basic" and "free" determines a vertex of the admissible set in n-space. A vertex is a point in the n-space where all of the inequalities are satisfied, *and* we are on the intersection of n-hyperplanes determined by setting n of the inequalities actually equal to 0. Each inequality is either of the form $x_i \geq 0$, or of the form $w_j = x_{n+j} = a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n - b_j \geq 0$. To get n-hyperplanes, we choose a subset of the variables $\{x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}\}$ to set equal to zero. This is the same as choosing the n basic variables, for these are the ones which we set equal to zero.

There is a problem with the above description of the vertices. After we choose n variables to call "basic", we can then solve for the values of the other m free variables using the constraint equations. For this to give a point in the feasibility set, we must have that all of the variables are non-negative! So after solving for the other m variables, we must throw the point out if it does not give the other variables ≥ 0.

To start the simplex method, we must have a vertex in the feasibility set. We find this vertex with trial and error. That is, start with some choice of variables to call "basic", then see if the remaining free variables turn out non-negative. If so, go on to the next step. But if not, then choose some other subset of n "basic" variables and try them out. It's not pretty, but this is the way to start the method. (See the worked examples at the end of the handout.)

EXAMPLE 3.1 If we start with the cost function in the variables $x_1, \ldots, x_n$ with the matrix of constraints from section 1, then when does the point $x_1 = 0, \ldots, x_n = 0$ lie in the admissible set? We saw above that for this value of the (basic) variables, the remaining (free) variables are given by $w_j = x_{n+j} = -b_j$. This must be ≥ 0 for each j. So to have the origin in the admissible set, we must have that all $-b_j \geq 0$ or that none of the range values b_j are positive.

4 Pivots and the simplex algorithm

The free variables for the tableau form of the simplex method are the variables corresponding to the "identity matrix" in the tableau. When we first set a problem up in tableau form, the variables $x_1, \ldots, x_n$ are usually set equal to zero, so they are basic, and the new variables $w_1, \ldots, w_m$ are the free variables. (They are introduced with a negative sign on the diagonal, bu this is easily fixed by multiplying the rows by -1.). This may or may not correspond to a vertex in the feasible set as we just discussed. The cost at this choice of free and basic variables is the negative of the entry in the lower right-hand corner of the tableau. What we need to do next is see if there is some better choice of free and basic variables which will result in a higher or lower cost. We will follow the text, and look for a minimum, so we are trying to lower the cost.

Each choice of n basic variables gives m equations in the remaining m "free" variables, so determines a point in the space of variables. The idea is to take some edge of the feasibility set, and see if the cost is less at the other end of the edge! A *point* corresponds to having m free variables in the m constraint equations. An *edge* will be given by adding one more free variable. This free variables are allowed to have non-zero (positive) values, so when we release one more variable to be free, the range of values trace out a line in space. So the point of the method is to consider all of the basic variables, which are being held fixed at zero, and determine which to release to become a free variable.

The other end of the line we are going to look at is another vertex, which means another choice of n variables which are "basic". The new vertex is obtained by *swapping* one of the free variables for one of the basic variables. This is what we next describe how to do.

To summarize the above discussion, we want to set "free" a basic variable, and choose a free variable and make it "basic" by setting it equal to zero. The names desribe whatthe variables are allowed to do!

Here are the remaining steps of the simplex method algorithm:

1) The first step in the method is to put the free variables (chosen in the previous step so that the corresponding vertex is in the feasibility set) in the front of the tableau. This means we make *column swaps* to get the m free variables up front.

2) Next, do row operations on the tableau to make the first m columns and m rows into the identity matrix. When we consider the basic variables as being equal to zero, this makes each free variable equal to

the value in the "b" column at the end of its row. Also, in this step, we must make the entries in the cost row, which is the very bottom, all equal to zero. (This correpsonds to substituting into the cost function the *negatives* of the values of these free variables, so at the lower right end of the cost row will appear the negative of the actual cost.)

3) We now must choose a basic variable to make free. Setting free a basic variable allows its value to "go positive", which will then change the cost function. *For a minimum* of the cost function, we look for the basic variable with the *most negative entry* on the cost row. This is the basic variable that we are going to set free. The negative cost entry associated with this entry means that when the variable goes positive, the cost will be driven down!

4) The next step is to choose a free variable to make basic. The free variables are those corresponding to the first m columns, are equally to the first m rows. (The last row is always the cost row.) To select which free variable to make basic, we look in the column of the basic variable chosen just before, and consider the *ratios* of the last entry in the row (in the "b" column) divided by the entry in the basic variable column. This will be some number, say we call it b'_i/a'_{ij}, where the "j" denotes that the basic variable is in the j^{th} column, and the "i" means that we are looking at the i^{th} row or i^{th} free variable. Choose the row which has the *smallest* ratio! (This is choosing the free variable which will fall the farthest when we make it basic; i.e., set it equal to zero. The net result is to drive the cost down fastest.)

5) We now have identifies the basic and the free variable that weare going to switch. The next step is simply to switch these two columns. This gives us a new set of free variables as the first m columns. We can go back to step 1), and proceed to make these columns look like the identity.

6) The procedure above repeats as often as there is a negative entry in the cost row. *Stop when there are no more negative entries.*

7) The entry in the lower right-hand corner represents the cost of the vertex corresponding to the chosen free variables, as long as the first m entries of the cost row have been row-reduced to zero. This is after step 2). When we stop in step 6) because there are no more negative entries, then the optimal lowest cost appears as the *negative* of the entry in the lower right-hand corner!

8) At each iteration of the simplex algorithm procedure, we have to switch two columns. We can label the columns when we start, and when you switch the two columns, switch the labels. Then when the algorithm stops, the labels at the top of the rows give the names of the free variables and we can then identify the vertex where the minimum takes place. If you don't care where the minimum is, but only want its value, then labeling is not necessary.

9) Finally, the simplex method works equally well to produce a maximum. To do this, look for the *most positive entry* in the cost row in step 3), and repeat the rest of the algorithm as above.

There are many refinements of the above method. As described, we do too many operations on the row entries. The modified simplex method takes this into account, and is more efficient. The other issue to address is what to do when there is no unique decision in the above procedure. That is, maybe there are several equally negative entries, etc. Then a computer algorithm must base the decision on what to do next based on additional data.