Linear Programming Redux

Jim Bremer
May 12, 2008

The purpose of these notes is to review the basics of linear programming and the simplex method in a clear, concise, and comprehensive way. The book contains all of this material, but it is unfortunately spread across several chapters and, in my opinion, confusing in part.

These notes are a bit more demanding then the book; if you can read and thouroughly understand them then you are doing very well in the course. The difficulty that most students will encounter in these notes is that they assume a thorough knowledge of linear algebra. I have said several times in class and will repeat now: there is, in my experience, almost no subject with a better reward/effort ratio than linear algebra (statistics being perhaps more rewarding still per unit of effort). A thorough, complete knowledge of elementary linear algebra will serve anyone in a technical field well indeed.

Note that this discussion is not comprehensive; in particular, I have omitted any discussion of numerical implementation and stability, as well as any discussion of the geometry underlying linear programs.

1. Underdetermined Systems

We begin by reviewing underdetermined systems of linear equations. Let A be an $m \times n$ matrix with $n > m$. Then the system of equations

$$Ax = b \tag{1}$$

is *underdetermined* — it has more variables than equations. Suppose we were to row reduce the augmented matrix $\left(\begin{array}{c|c} A & b \end{array} \right)$ in order to solve the system (1). Because the matrix A has more columns than rows we will necessarily have free variables (if this isn't clear to you, work out an example now). So we cannot expect a unique solution for this system.

Of course, it is also possible that (1) *has no solutions at all* (if this isn't clear, stop and find an example). By making an additional assumption on the matrix A, we can ensure that the system (1) has a solution for every $b \in \mathbb{R}^m$. In particular, we will generally assume that the matrix A has *rank* m.

Recall that the rank of a matrix is the dimension of its column and row spaces (these two dimensions are equal). So the assumption that A has rank m means that there are no redundant equations in the system (1), or, equivalently, that the column space of A must be all of $\mathbb{R}^m$ — in other words, for every b there is some solution of the system (1). It is also equivalent to saying that some set of m columns of the matrix A forms a basis for $\mathbb{R}^m$.

We will be particularly interested in a distinguished class of solutions of (1). We say that $x \in \mathbb{R}^n$ is a *basic solution* for the system (1) if there is a set of indices $\{i_1, \ldots, i_m\} \subset \{1, 2, \ldots, n\}$ such that:

1. $x_j = 0$ for $j \notin \{i_1, \ldots, i_m\}$, and
2. the corresponding columns $A_{i_1}, \ldots, A_{i_m}$ of the constraint matrix A form a basis for $\mathbb{R}^m$.

Note that the columns of the matrix corresponding to a basic solution form a submatrix B of A which is invertible; indeed, a set of m vectors $v_1, \ldots, v_m$ in $\mathbb{R}^m$ form a basis if and only if the matrix whose columns are $v_1, \ldots, v_m$ is invertible (check this!).

Basic solutions for (1) are (relatively) easy to find: simply find a submatrix B consisting of the columns $A_{i_1}, \ldots, A_{i_m}$ of A which form a basis (perhaps via Gram-Schmidt Orthogonalization), solve the $m \times m$ system $Bz = b$ for z, and form the basic solution x with entries

$$x_j = \begin{cases} z_{i_j} & \text{if } j \in \{i_1, \ldots, i_m\} \\ 0 & \text{otherwise.} \end{cases}$$

Since the matrix B is invertible, there is one and only one basic solution x associated with the columns $i_1, \ldots, i_m$. We will call that solution *the* basic solution associated with the columns $i_1, \ldots, i_m$. Alternately, we will say that x is the basic solution corresponding to the invertible submatrix B.

These facts are sufficiently important that we will repeat them in Lemma form; we have proved the following:

LEMMA 1.1. *Suppose that A is an $m \times n$ matrix, $n \geq m$, of rank m and further suppose that the columns $i_1, \ldots, i_m$ of A form a basis for $\mathbb{R}^m$. Then there is a unique vector x, which we will call the basic vector for the columns $i_1, \ldots, i_m$, such that*

1. $Ax = b$
2. $x_j = 0$ *for all* $j \notin \{i_1, \ldots, i_m\}$.

2. Linear Programs

A *linear program* is any optimization problem of the form

$$\begin{aligned} \text{max:} \quad & c^t x \\ \text{subject to:} \quad & Ax = b \\ & x \geq 0, \end{aligned} \tag{2}$$

where A is an $m \times n$ matrix, $n > m$, of rank m, c is a given row vector of length n, b is a given row vector of length m, and x is a row vector of unknowns of length n.

Remark 2.1. *Note that the assumption that A has rank m ensures that there are no redundant constraints in (2). It differs from the usual definition of linear program only in that it excludes certain infeasible problems from consideration; e.g., the problem*

$$\begin{aligned} \textit{max:} \quad & x_1 + x_2 \\ \textit{subject to:} \quad & x_1 - x_2 = -1 \\ & -x_1 + x_2 = -1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

is excluded.

We will call a vector x which satisfies the constraints $Ax = b$ and $x \geq 0$ a *feasible* vector for the program (2). Moreover, a feasible vector x for which cx obtains a maximum among the set of all feasible vectors is called an *optimal feasible vector* for the program (2), or, more simply, a *solution* to (2). The linear function cx is called the *objective* function for the program (2).

The first thing we should note about the problem (2) is that the constraint equations $Ax = b$ are underdetermined. Because of our assumption about the rank of A, we expect there to be an infinite number of solutions for this system of equations. That only stands to reason, since the problem is asking us to find from among the infinite number of $x's$ satisfying the constraints, a single x such that cx is maximized.

Our second immediate observation is that, despite the fact that the system $Ax = b$ has an infinite number of solutions, it is not necessarily true that there exists even a single x satisfying all of the constraints. For example, this is the case for linear program

$$\begin{aligned} \text{max:} \quad & x_1 + x_2 + x_3 \\ \text{subject to:} \quad & x_1 + x_3 = -1 \\ & x_2 - x_3 = -1 \\ & x_1, x_2, x_3 \geq 0. \end{aligned} \tag{3}$$

We call a linear program for which there are no vectors x satisfying both $Ax = b$ and $x \geq 0$ *infeasible.*

Clearly, infeasible linear programs do not admit a solution. There is another way in which a linear program can fail to have a solution. There might not be a maximum value of the objective function cx. In such cases, we say that the program (2) is *unbounded.*

We close this section with a final observation about (2): if the set of x which satisfy the constraints is bounded (as a set in $\mathbb{R}^n$), then the problem (2) cannot be unbounded. As with all sufficiently elementary facts, there are many different incantations we can invoke to see that this is so (e.g., a continuous function on a compact set in $\mathbb{R}^m$ obtains its maximum). This implies, for example, that a linear program of the special form

$$\begin{aligned} \text{max:} \quad & c^t x \\ \text{subject to:} \quad & Ax = b \\ & l \le x \le u \end{aligned}$$

cannot be unbounded (it can, however, still be infeasible).

3. The Fundamental Theorem

The Fundamental Theorem of Linear Programming (for which there is a separate handout on the website including a proof) is an extremely powerful statement about solutions of problems of the form (2).

THEOREM 3.1. *(FTLP) Suppose that A is an $m \times n$ matrix, $n > m$, of rank m. Then, if the linear program*

$$\begin{aligned} \textit{max:} \quad & c^t x \\ \textit{subject to:} \quad & Ax = b \\ & x \ge 0 \end{aligned} \tag{4}$$

admits a solution, there is a solution $x \in \mathbb{R}^m$ and a set $\{i_1, \ldots, i_m\}$ of m indices such that the following properties:

1. $x_j = 0$ for $j \notin \{i_1, \ldots, i_m\}$,
2. the corresponding columns $A_{i_1}, \ldots, A_{i_m}$ of the constraint matrix A form a basis for $\mathbb{R}^m$.

In other words, if there is a solution to (4), then there is a solution to (4) which is a basic solution for the constraint equation $Ax = b$.

Solutions of this form are sufficiently important that we immediately make a definition: a vector x which satisfies the constraints $Ax = b$ and $x \ge 0$ and for which, in addition, the conditions (1) and (2) above hold will be called a *basic feasible vector* for the program (4). If, in addition, x is a solution to (4), we will call x a *basic solution* for the linear program (note the distinction between a solution to the constraint equations and a solution the linear program as a whole).

The Fundamental Theorem immediately reduces the problem of finding a solution to a linear program from a search through a potentially infinite set to a search through a finite one. In particular, it suggests that instead of looking for x which satisfy the constraints, we should look for subsets of columns of the constraint matrix A which form bases.

We can now describe our first algorithm for solving the linear program (4). To simplify things, we will assume that the problem under consideration is not unbounded. Note that it is not difficult to modify this algorithm to detect unboundedness, but we won't do that here, because we will be shortly introducing a superior algorithm. We begin the algorithm by letting $\eta = -\infty$ and $x_0 = 0$ and then executing the following sequence of steps for each set of m indices $i_1, \ldots, i_m$ in the set $\{1, \ldots, n\}$:

Step 1. Form the matrix

$$B = \begin{pmatrix} A_{i_1} & A_{i_2} & \ldots & A_{i_m} \end{pmatrix}$$

of the columns of A corresponding to the indices $i_1, \ldots, i_m$.

Step 2. If B is invertible (that is, if the columns of B are a basis for $\mathbb{R}^m$), then find the unique solution $z \in \mathbb{R}^m$ to the system of equations $Bz = b$. Otherwise, we are finished processing this set of indices.

Step 3. If z satisfies the constraints $z \ge 0$, then we have found a basic feasible vector x, which is defined by

$$x_j = \begin{cases} z_{i_j} & \text{if } j \in \{i_1, \ldots, i_m\} \\ 0 & \text{otherwise.} \end{cases}$$

Otherwise, are finished processing this set of indices.

Step 4. If $c^t x > \eta$, then we let $x_0 = x$ and $\eta = cx$.

This procedure will terminate in a finite number steps since there are

$$\begin{pmatrix} n \\ m \end{pmatrix} = \frac{n!}{(n-m)!m!}$$

ways of choosing m indices from a set of n possible indices, and upon termination η will be either be $-\infty$, in which case the problem is infeasible, or η will be the maximum value of the objective function and x_0 will be a basic solution.

4. Tableaus

Consider the linear program

$$\text{(5)} \qquad \begin{aligned} \text{max:} \quad & c^t x \\ \text{subject to:} \quad & Ax = b \\ & x \geq 0, \end{aligned}$$

where A is an $m \times n$ matrix, $n > m$, of rank m, and suppose that $\tilde{x}$ is a basic solution to the constraint equation $Ax = b$. Note: we are not assuming that it is a solution to the entire linear program, just the constraint equation; indeed, we are not even assuming that it is feasible.

In the future, we will call such vectors *basic vectors* for the linear program (5) to avoid any confusion over the word "solution." A basic vector which is feasible will be called, of course, a *basic feasible vector* and a basic vector which is feasible and optimal will be called a *basic solution* for the LP.

We can, by rearranging the columns of A and the rows of x, ensure that the basis associated with $\tilde{x}$ consists of the first m columns of A. We can then write the LP (5) as

$$\text{(6)} \qquad \begin{aligned} \text{max:} \quad & c_1^t x_B + c_2^t x_N \\ \text{subject to:} \quad & \begin{pmatrix} B & N \end{pmatrix} \begin{pmatrix} x_B \\ x_N \end{pmatrix} = b \\ & x_B, x_N \geq 0, \end{aligned}$$

where we have partitioned the variables $x_1, \ldots, x_n$ into two sets: the basic variables x_B corresponding to the first m columns of A and the nonbasic variables x_N. This leads to a corresponding partitioning of the constraint matrix A into the $m \times m$ invertible matrix B and the $m \times (n - m)$ matrix N.

Remember that we started with some *basic* vector $\tilde{x}$ for the constraint equations $Ax = b$. We can also partition its entries. We will let $\tilde{x_B}$ denote the values of the basic entries of $\tilde{x}$ and $\tilde{x_N}$ denote the nonbasic entries. Because $\tilde{x}$ is a basic solution, its other entries are zero; i.e., $\tilde{x_N} = 0$.

The constraint equation in (6) is of the form

$$Bx_B + Nx_N = b.$$

Since B is invertible, we can multiply both sides of this equation by B^{-1}, which yields

$$\text{(7)} \qquad x_B + B^{-1}Nx_N = B^{-1}b.$$

We will make two observations about the equation (7). First, plugging the vector $\tilde{x}$ into (7), we get:

$$\tilde{x_B} + B^{-1}N\tilde{x_N} = B^{-1}b,$$

or (since $\tilde{x_N} = 0$),

$$B^{-1}b = \tilde{x_B}.$$

Moreover, (7) allows us to rewrite the objective function in the form

$$\begin{aligned} cx &= c_1^t x_B + c_2^t x_N \\ &= c_1^t(\tilde{x_B} - B^{-1}N)x_N + c_2^t x_N \\ &= \tilde{\xi} + \tilde{c}^t x_N \end{aligned}$$

where $\tilde{c}$ is a column vector of length $n - m$.

Thus we can rewrite the LP (5) as

$$\text{(8)} \qquad \begin{aligned} \text{max:} \quad & \tilde{\xi} + \tilde{c}^t x_N \\ \text{subject to:} \quad & \begin{pmatrix} I & B^{-1}N \end{pmatrix} \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \tilde{x_B} \\ & x_B, x_N \geq 0, \end{aligned}$$

We will call the form of the linear program (8) the *tableau* associated with the basic vector $\tilde{x}$. We say that (8) is a *feasible tableau* if $\tilde{x_B} \geq 0$ and it is an *optimal tableau* if $\tilde{x}$ is an optimal feasible vector.

This form has several useful properties:

1. It explicitly exhibits the value of $\tilde{x}$ in the constants that appear on the right hand side of the constraint equation.
2. It is very easy to determine if $\tilde{x}$ is a feasible vector for the linear program: if $\tilde{x_B} \geq 0$, then $\tilde{x}$ is feasible, otherwise it is not.
3. Assuming $\tilde{x}$ is feasible, it is also easy to determine if $\tilde{x}$ is a solution to the LP. Although this form of the LP is associated with a particular vector $\tilde{x}$, it is, in fact, equivalent to the original LP (5). Thus, we can test the feasibility of any vector x and evaluate the objective function cx using (8). In particular, we examine the vector $\tilde{c}$. If none of the entries of $\tilde{c}$ are positive, then there can be no other basic feasible vector x for which cx takes on a value larger than $\tilde{\xi}$ (note the feasibility assumption is important here because it means the entries of x_N we are testing are nonnegative).

Remark 4.1. *We are used to writing down the tableau form of a linear program in a table in the following manner:*

nonbasic	*basic*	
$B^{-1}N$	I_m	$\tilde{x_B}$
$-\tilde{c}$	*0*	$\tilde{\xi}$

5. The Simplex Method: Overview

In this section we introduce the simplex method, an improved algorithm for solving linear programs

$$\begin{aligned} \text{max:}\quad & c^t x \\ \text{subject to:}\quad & Ax = b \\ & x \geq 0, \end{aligned} \tag{9}$$

where A is an $m \times n$ matrix, $n > m$, of rank m.

Suppose that $\tilde{x}$ and $\tilde{y}$ are basic vectors for the LP (9), and further suppose that $\tilde{X}$ is associated with the columns $\{i_1, \ldots, i_m\}$ of A and $\tilde{y}$ with the columns $\{j_1, \ldots, j_m\}$. Then we say that $\tilde{x}$ and $\tilde{y}$ are *adjacent* if the sets $\{i_1, \ldots, i_m\}$ and $\{j_1, \ldots, j_m\}$ have $m - 1$ elements in common. In other words, two basic vectors $\tilde{x}$ and $\tilde{y}$ are adjacent if their associated bases differ by one element.

The idea behind the simplex method is quite simple: it is an iterative method, which starting with an initial basic feasible vector, moves from one basic feasible vector to another adjacent one in an effort to increase the value of the objective function. For each iteration j, we form the tableau

$$\begin{aligned} \text{max:}\quad & \tilde{\xi_j} + \tilde{c_j}^t x_N \\ \text{subject to:}\quad & \begin{pmatrix} I & B^{-1}N \end{pmatrix} \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \tilde{x_j} \\ & x_B, x_N \geq 0, \end{aligned} \tag{10}$$

for the associated basic feasible vector $\tilde{x_j}$ (there is a slight abuse of notation here: we are identifying the basic part of $\tilde{x_j}$ with $\tilde{x_j}$). If $\tilde{x_j}$ is a solution for the linear program, it is evident from the coefficients of $\tilde{c_j}$ (see the observations about the tableaus above). Otherwise, we can choose a nonbasic variable in x_N whose coefficient is positive, called the entering variable, which we will move into the basis. Of course, we must swap this variable with a properly chosen basic variable from x_B, called the leaving variable, in order to maintain a basis. In most cases, this results in an increase in the objective function.

Remark 5.1. *It is very important to understand what having an initial basic feasible vector entails. It means not only do we have a vector $\tilde{x}$ with no more than m nonzero entries which is feasible ($\tilde{x} \geq 0$ and $A\tilde{x} = b$), but also that the associated columns of the constraint matrix A form a basis. Not any old basis of columns of A will do — we need a basis consisting of columns of A such that the associated basic solution is nonnegative!*

6. A Simplex Method Step

In this section, we describe in detail what a single step of the simplex method entails. First, we assume that the underlying linear program is of the form

$$\begin{aligned} \text{max:} \quad & c^t x \\ \text{subject to:} \quad & Ax = b \\ & x \geq 0, \end{aligned} \tag{11}$$

where A is an $m \times n$ matrix, $n > m$, of rank m. Next, we suppose that at the j^{th} iteration we have the tableau

$$\begin{aligned} \text{max:} \quad & \tilde{\xi} + \tilde{c}^t x_N \\ \text{subject to:} \quad & \begin{pmatrix} I & B^{-1}N \end{pmatrix} \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \tilde{x_B} \\ & x_B, x_N \geq 0, \end{aligned} \tag{12}$$

which is associated with the basic feasible vector $\tilde{x}$. Throughout this section, whenever we have a vector $x \in \mathbb{R}^n$, we will denote by x_B the part of x corresponding to the basic variables for the tableau (12) and by x_N the part of x corresponding to the nonbasic variables for this tableau.

Let us be explicit about the dimensions of the various blocks of the constraint matrix in (12): the matrix B is an $m \times m$ matrix, the idenity block in (12) is also $m \times m$, and $B^{-1}N$ is $m \times (n - m)$. This, of course, means that x_B represents the m basic variables and x_N represents the $n - m$ nonbasic variables.

Our goal is to find an adjacent basic feasible vector $\hat{x}$ for which the objective function is larger than $\hat{x}$. Recall that a basic vector adjacent to $\tilde{x}$ is a basic vector which shares $m - 1$ of its basic columns from the matrix A in common with $\tilde{x}$. So, in other words, we are going to proceed by switching one basic variable with a nonbasic variable.

We first choose the nonbasic variable from x_N which will become basic in the next iteration. We do this by examining the coefficients of $\tilde{c}$ and looking for one which is positive. Increasing the corresponding nonbasic variable will increase the objective function (note that if no such nonbasic variable can be found, then $\tilde{x}$ is already optimal). We call the chosen nonbasic variable the *entering variable* since it will be entering the basis for the next iteration.

Of course, when we increase the entering variable, we will need to adjust all of the basic variables in order to ensure that the constraint equation $A\hat{x} = b$ is still satisfied. We are proposing to change the nonbasic part of $\tilde{x}$ by letting

$$\hat{x_N} = te_k,$$

where t is a suitably chosen positive real number and e_k is the standard basis vector corresponding to the nonbasic variable we have chosen (i.e., it has a one in the correct position and zeros everywhere else). Since the equation

$$\hat{x_B} + B^{-1}N\hat{x_N} = x_B \tag{13}$$

must hold for any solution of the constraint equations, we have

$$\hat{x_B} = x_B - tB^{-1}Ne_k.$$

We define the vector Δx by

$$\Delta x = B^{-1}Ne_k.$$

This is the *direction* of the change in $\hat{x_B}$. Notice that Ne_k is just the column of the original constraint matrix A which corresponds to the entering variable.

We would like to increase t as much as possible — recall that the objective function will increase by $\tilde{c}_k t$. The value of t is usually constrained, however, by the requirement that $\hat{x_B} \geq 0$. In particular, when the value of coordinate i of Δx is greater than 0, then the corresponding equation in (13) gives us a bound on t. So we let t be

$$\min_i \frac{(x_B)_i}{(\Delta x)_i},$$

where the minimum ranges over all indices i for which $(\Delta x)_i$ is positive.

If there are no such indices, then the problem is unbounded. That is, we can increase t to infinity without violating the constraints, and as we do so, the objective function will increase without bound.

That is, we increase t until one or more of the variables in $\hat{x_B}$ becomes zero. We now choose from among the set of $\hat{x_B}$ a *leaving variable*. That is, a variable to move out of the basic set x_B and into the nonbasic set x_N at the next iteration.

We have now computed the value of the basic feasible vector at the next iteration: $\hat{x_B}$ and $\hat{x_N}$. It only remains to do bookkeeping: to update the list of variables that are basic and nonbasic for the next iteration.

Remark 6.1. *Note that it is never necessary while performing the simplex method algorithm to actually compute the form of the constraint matrix which appears in (12). Indeed, it is more convenient to leave the constraint matrix in its original form (11), and compute the direction Δx by solving a system of equations.*

Remark 6.2. *Assuming that we do not run into unboundedness, the new set of basic variables do indeed correspond to a set columns of A which forms a basis. To see this, note that the vector Δx records the coefficients of the entering column with respect to the current basis. Because the entry of Δx cooresponding to the leaving variable is nonzero (by definition), it means that the entering column is not in the span of the basis columns excluding the leaving column. This implies the the resulting set of columns is a basis.*

7. Degeneracy

8. Initialization I

There is an obvious unresolved difficulty with the simplex method: we need to find a basic feasible vector in order to start the simplex method (if we start with an infeasible vector, we might never find a solution to the LP). As was remarked upon above, this is a nontrivial task which entails more than finding an invertible submatrix of the constraint matrix A.

We have been spoiled by the textbook's practice of writing linear programs in the form:

$$\begin{aligned} \text{max:} \quad & c^t x \\ \text{subject to:} \quad & Ax \leq b \\ & x \geq 0. \end{aligned} \tag{14}$$

This form is particularly nice for initialization, because once m slack variables have been introduced (14) takes on the form

$$\begin{aligned} \text{max:} \quad & c^t x \\ \text{subject to:} \quad & \begin{pmatrix} A & I \end{pmatrix} \begin{pmatrix} x \\ w \end{pmatrix} = b \\ & x, w \geq 0. \end{aligned}$$

This makes it trivial to find an initial basis for the constraint matrix (we simply pick the identity submatrix I). Of course, the resulting basic vector is feasible only if $b \geq 0$, so there is still work to be done even in this case, but nonetheless it is easier than the initialization of a general LP.

Instead of considering this simple case, we will once again the consider the general LP

$$\begin{aligned} \text{max:} \quad & c^t x \\ \text{subject to:} \quad & Ax = b \\ & x \geq 0, \end{aligned} \tag{15}$$

where A is an $m \times n$ matrix, $n > m$, of rank m. Without loss of generality, we may assume that $b \geq 0$ (if $b_i < 0$ for some i then we can multiply the corresponding constraint by -1). In order to initialize (15), we introduce the auxiliary linear program

$$\begin{aligned} \text{min:} \quad & y_1 + y_2 + \ldots + y_m \\ \text{subject to:} \quad & Ax + y = b \\ & x, y \geq 0. \end{aligned} \tag{16}$$

It should be clear that the original problem (15) is feasible if and only if (16) has a solution such that

$$y_1 = y_2 = \ldots = y_m = 0.$$

Moreover, it is easy to find an initial feasible vector for (16): we simply let $y = b$ and $x = 0$. Then the corresponding submatrix of the constraint matrix is the identity (and so clearly invertible) and because b can be assumed to satisfy $b \geq 0$, this vector is feasible.

Thus we initialize the simplex method by attempting to solve the auxiliary LP (16). If, for the resulting solution (x, y), we have $y_1 = y_2 = \ldots = y_n = 0$, then we have found a basic feasible vector for the original LP.

There is one complication, however. Finding a basic solution of (16) such that (8) holds means that we have found a basic feasible vector for (15), but it does not mean that we can necessarily identify the basis! In particular, if all of the basic variables for the solution of (16) are $x's$ then it is obvious what columns of A to choose: if the basic variables for the auxiliary solution are

$$x_{i_1}, x_{i_2}, \ldots, x_{i_m}$$

then the columns $A_{i_1}, \ldots, A_{i_m}$ are a basis for $\mathbb{R}^m$ and the associated solution of $Ax = b$ is feasible. However, if one or more of the y_j are included in the basic variables for the solution of (16), then more work must be done to find a basis of columns of A.

We could finish the initialization process by completing our basis with a set of additional columns from A, but in most cases, it is easier and more elegant to simply use a foolproof initialization scheme that we will discuss below.

Remark 8.1. *Note that we can only have a basic solution for (16) with one or more y_j as basic variables if the solution has fewer than m nonzero entries.*

9. Duality

The dual of the linear program

$$\begin{aligned} \text{max: } & c^t x \\ \text{subject to: } & Ax \le b \\ & x \ge 0, \end{aligned} \tag{17}$$

is the linear program

$$\begin{aligned} \text{min: } & b^t y \\ \text{subject to: } & A^t y \ge c \\ & y \ge 0. \end{aligned} \tag{18}$$

It is easy to see that the dual of the dual (18) program is again the original problem (17), which we refer to as the *primal* problem. This definition is motivated by the search for an upper bound for the objective function of (17) using the constraint equations. Your book does a good job of explaining this in the beginning of Chapter 5. Note that there is one constraint in the dual problem for every variable in the primal problem and one variable in the dual for each constraint in the primal.

Since (17) is not the usual form for linear programs, our first task is to find the dual for a program in the usual form:

$$\begin{aligned} \text{max: } & c^t x \\ \text{subject to: } & Ax = b \\ & x \ge 0. \end{aligned} \tag{19}$$

We proceed by rewriting the constraints in (19) as inequalities:

$$\begin{aligned} \text{max: } & c^t x \\ \text{subject to: } & Ax \le b \\ & -Ax \le -b \\ & x \ge 0. \end{aligned} \tag{20}$$

It is now clear that the dual of (19) is the linear program

$$\begin{aligned} \text{min: } & b^t y_1 - b^t y_2 \\ \text{subject to: } & \begin{pmatrix} A^t & -A^t \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \ge b \\ & y_1, y_2 \ge 0. \end{aligned} \tag{21}$$

Letting $z = y_1 - y_2$, we see that (21) is equivalent to

$$\begin{aligned} \text{min:} \quad & b^t z \\ \text{subject to:} \quad & A^t z \geq c \\ & z \text{ free.} \end{aligned} \tag{22}$$

In other words, we can combine the two sets of dual variables y_1 and y_2 we got from splitting the equality in the primal into a single dual variable z which is not constrained to be nonnegative.

In general we will find this to be the case: an equality constraint in the primal corresponds to a free variable in the dual and a free variable in the primal corresponds to an equality constraint in the dual.

We are now in a position to prove the Weak Duality Theorem, which confirms the intuition which lead to the definition of the dual:

THEOREM 9.1. *(Weak Duality). If x is a feasible vector for the primal linear program (17) and y is a feasible vector for its dual problem (18), then $c^t x \leq b^t y$.*

Proof: Using the dual constraint equation $A^t y \geq c$ and the primal constraint equation $Ax \leq b$ we have:

$$c^t x \leq (y^t A)x = y^t(Ax) \leq y^t b = b^t y.$$

QED.

Finally, we close this section with a discussion of affine objective functions, which will prove useful in the next section. Consider the optimization problem

$$\begin{aligned} \text{max:} \quad & \xi + c^t x \\ \text{subject to:} \quad & Ax \leq b \\ & x \geq 0, \end{aligned} \tag{23}$$

where A a is an $m \times n$ matrix, $n > m$, of rank m. Notice that the objective function of (23) is not a linear function, but rather an *affine* function. It should be clear that this is no obstacle to the solution of (23). Indeed, we can write (23) as a linear program in the form:

$$\begin{aligned} \text{max:} \quad & c^t x + z \\ \text{subject to:} \quad & Ax \leq b \\ & z = \xi \\ & x \geq 0, \text{z free.} \end{aligned} \tag{24}$$

By a slight abuse of notation, we will refer to problems of the form (23) with affine objective function as linear programs.

The dual of the linear program (24) can then be written as:

$$\begin{aligned} \text{min:} \quad & b^t y + \xi w \\ \text{subject to:} \quad & A^t y \geq c \\ & w = 1 \\ & y \geq 0, \text{w free.} \end{aligned} \tag{25}$$

This linear program is evidently equivlent to the optmization problem:

$$\begin{aligned} \text{min:} \quad & \xi + b^t y \\ \text{subject to:} \quad & A^t y \geq c \\ & y \geq 0. \end{aligned} \tag{26}$$

So we see that the dual of the program (23) with an affine objective function is the linear program (26) which also has an affine objective function.

10. The Dual Tableau

The primary observation of this section — and the most important single fact in the theory of duality — is that the dual of a tableau for the primal problem is a itself a tableau for the dual problem. This fact and the form of the two corresponding tableaus imply all sorts of nice results (e.g., strong duality).

To see that this is so, suppose that

$$\begin{aligned} \text{max:}\quad & \tilde{\xi} + \tilde{c}^t x_N \\ \text{subject to:}\quad & \begin{pmatrix} I & B^{-1}N \end{pmatrix} \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \tilde{x} \\ & x_B, x_N \geq 0, \end{aligned} \tag{27}$$

is a tableau associated with a basic feasible vector $\tilde{x}$ for a primal problem. Using what we learned in the last section (check this!), we find that the dual of (27) is

$$\begin{aligned} \text{min:}\quad & \tilde{\xi} + \tilde{c}^t y \\ \text{subject to:}\quad & \begin{pmatrix} I \\ (B^{-1}N)^t \end{pmatrix} y \geq \begin{pmatrix} 0 \\ \tilde{c} \end{pmatrix} \\ & y \text{ free,} \end{aligned} \tag{28}$$

Of course, we can rewrite this as

$$\begin{aligned} \text{min:}\quad & \tilde{\xi} + \tilde{x}^t y \\ \text{subject to:}\quad & (B^{-1}N)^t y \geq \tilde{c} \\ & y \geq 0. \end{aligned} \tag{29}$$

If we introduce slack variables y_B into (29), and rename the variables already present y_N, then we arrive at the linear program

$$\begin{aligned} \text{min:}\quad & \tilde{\xi} + \tilde{x}^t y_B \\ \text{subject to:}\quad & \begin{pmatrix} I & -(B^{-1}N)^t \end{pmatrix} \begin{pmatrix} y_N \\ y_B \end{pmatrix} = -\tilde{c} \\ & y_B, y_N \geq 0. \end{aligned} \tag{30}$$

We call this linear program the *dual tableau* corresponding to the primal tableau (27). We can immediately make the following observations:

1. There is a correspondence between the dual variables y_B and the primal basic variables x_B, and between the dual variables y_N and the primal variables x_N.
2. As written, the basic variables for the tableau (30) are the y_N and **not** the y_B! So we have chosen to emphasize the relationship mentioned above rather than emphasize which variables are basic and which are nonbasic for the dual.
3. There is clearly a basic vector $\tilde{y}$ associated with the dual tableau: $y_B = 0$ and $y_N = \tilde{c}$.
4. The basic vector $\tilde{y}$ is not necessary feasible for the dual — we do not expect $\tilde{c} \geq 0$.

We will call the basic vector $\tilde{y}$ associated with the dual tableau (30) the *dual* basic vector of $\tilde{x}$. This is a very important idea: we associate with each basic vector of the primal a basic vector of the dual.

We close this section by proving the Strong Duality Theorem. Thanks to our discussion of tableaus and dual tableaus, the proof is trivial.

THEOREM 10.1. *(Strong Duality) Suppose that $\tilde{x}$ is a basic solution for the primal problem (e.g., a basic optimal feasible vector)*

$$\begin{aligned} \textit{max:}\quad & c^t x \\ \textit{subject to:}\quad & Ax \leq b \\ & x \geq 0. \end{aligned} \tag{31}$$

Then the corresponding basic vector $\tilde{y}$ for the dual

(32)
$$\begin{aligned} \textit{min:}\quad & b^t y \\ \textit{subject to:}\quad & Ax \geq c \\ & y \geq 0. \end{aligned}$$

of (31) is a basic solution (e.g., a basic optimal feasible vector for the dual). Moreover,

(33)
$$c^t \tilde{x} = b^t \tilde{y}.$$

Proof: Write the tableau

(34)
$$\begin{aligned} \text{max:}\quad & \tilde{\xi} + \tilde{c}^t x_N \\ \text{subject to:}\quad & \begin{pmatrix} I & B^{-1}N \end{pmatrix} \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \tilde{x_B} \\ & x_B, x_N \geq 0, \end{aligned}$$

for the basic vector $\tilde{x}$. That $\tilde{x}$ is feasible means that $x_B \geq 0$ and that it is optimal means that $\tilde{c} \leq 0$. Now the tableau

(35)
$$\begin{aligned} \text{min:}\quad & \tilde{\xi} + \tilde{x_B}^t y_B \\ \text{subject to:}\quad & \begin{pmatrix} I & -(B^{-1}N)^t \end{pmatrix} \begin{pmatrix} y_N \\ y_B \end{pmatrix} = -\tilde{c} \\ & y_B, y_N \geq 0. \end{aligned}$$

is the corresponding dual tableau — the one associated with the basic vector $\tilde{y}$. That $\tilde{c} \leq 0$, means that $\tilde{y}$ is feasible and $\tilde{x_B} \geq 0$ implies that it is optimal (since the dual is a *minization* problem). So $\tilde{y}$ is a basic optimal vector for the dual. Equation (33) now follows because we have $y_B = 0$ and $x_N = 0$ for the pair of vectors $\tilde{x}$ and $\tilde{y}$.
QED.

11. Initialization II

We now discuss the use of the dual problem to effect initialization.

12. Problems

1. Write down a feasible, bounded linear program and solve it.
2. What is the dual of the linear program

$$\begin{aligned} \text{max:}\quad & c^t x \\ \text{subject to:}\quad & Ax = b \\ & l \leq x \leq u? \end{aligned}$$

   Is it possible for the dual of this LP to be infeasible? What does that say about the primal, assuming it is feasible?
3. Why does the simplex method have to start at a basic feasible vector? Can we begin with a nonbasic feasible vector? How about an infeasible basic vector?
4. How could you go about making the first initialization procedure foolproof?
5. What is the dual of the linear program

$$\begin{aligned} \text{max:}\quad & c^t x \\ \text{subject to:}\quad & Ax = b \\ & x \text{ free ?} \end{aligned}$$

6. Show that the proof in Section 9 of the fact that the dual of a program with an affine objective function has an affine objective function still holds if the equality constraint $z = 1$ is changed to $z \leq 1$.

7. Consider the first algorithm for solving linear programs, which was introduced in Section 3. This algorithm can be easily modified to solve problems given in the form

$$\begin{aligned} \text{max:} \quad & c^t x \\ \text{subject to:} \quad & Ax = b \\ & x \text{ free} \ , \end{aligned}$$

where x is not constrained to be nonnegative. (It should be clear that any linear program can be written in this form — if not check it!!!) Indeed, the modified algorithm is clearly **simpler** than the algorithm as written.

Why is it difficult to modify the simplex method to operate on programs written in this form?