LINEAR PROGRAMMING AND EXTENSIONS

R. Chandrasekaran
The University of Texas at Dallas

August 28, 1998

0.1 INTRODUCTION:

Problem 1 *The following problem is known as the Linear Programming Problem or simply as the Linear Program:*

$$\min[\sum_{j=1}^{n} c_j x_j : \sum_{j=1}^{n} a_{i,j} x_j = b_i; 1 \leq i \leq m; x_j \geq 0; 1 \leq j \leq n]$$

The same problem can also be stated in matrix form as:

$$\min[cx : Ax = b; x \geq 0]$$

where A is a $m \times n$ matrix, b is a m-vector, and c and x are n-vectors. The decision variables in the problem are the $\{x_j\}$. This particular form of the linear program is called a standard form. It is so called because any linear program can be written in this form. There are other standard forms that are useful in other contexts; this particular one is useful in the context of the simplex algorithm. We now state the other forms:

Problem 2 *The inequality form is:*

$$\min \sum_{j=1}^{n} c_j x_j : \sum_{j=1}^{n} a_{i,j} x_j \geq b_i; 1 \leq i \leq m; x_j \geq 0; 1 \leq j \leq n$$

which is matrix form looks like:

$$\min cx : Ax \geq b; x \geq 0$$

There is another standard inequality form:

$$\max \sum_{j=1}^{n} c_j x_j : \sum_{j=1}^{n} a_{i,j} x_j \leq b_i; 1 \leq i \leq m; x_j \geq 0; 1 \leq j \leq n$$

which is matrix form looks like:

$$\max cx : Ax \leq b; x \geq 0$$

Problem 3 *The most general for7m of a linear program has three kinds of constraints: equations, inequalities of the type $\leq$; inequalities of the type $\geq$. It has three kinds of variables: those that are required to be ≥ 0; those that are required to be ≤ 0; and those that can take on positive, negative or zero value – these are called unrestricted (in sign) variables. There are two possibilities for the objective function: maximize or minimize. Thus the general LP may look like the following in matrix form:*

$$\begin{gathered} \min(\max) c^1 x^1 + c^2 x^2 + c^3 x^3 \\ A^{11} x^1 + A^{12} x^2 + A^{13} x^3 \leq b^1 \\ A^{21} x^1 + A^{22} x^2 + A^{23} x^3 \geq b^2 \\ A^{31} x^1 + A^{32} x^2 + A^{33} x^3 = b^3 \\ x^1 \geq 0; x^2 \leq 0; x^3 \textit{ unrestricted} \end{gathered}$$

Exercise Show how to convert each one of these forms to the other.

Please note that there is no restriction on the number of variables or the number of constraints (as they long as they are finite!). Of course we expect that the solution of a larger problem will take more time.

What is not allowed: Strict inequalities such as $<$ or $>$; requiring that the variables take on only integer values. (the latter is really a nonlinear condition of the type $\sin(\pi x_j) = 0$).

Redundant equations:

If the set of solutions is unaltered by dropping an equation from (adding to) a system, then this equation is said to be a redundant equation (a similar statement applies to any constraint). In particular, if $a_{i,j} = \sum_{k \neq i} \lambda_k a_{k,j}; b_i = \sum_{k \neq i} \lambda_k b_k$, then the i^{th} equation is clearly redundant. Indeed all redundant equations are of this type. Very often we will assume that there are no such equations is our system; we also provide means to test if this is the case or not.

Pivot Operation:

This is the central operation in the simplex algorithm. It is equivalent to solving for some variables in terms of others. It is done with respect to a nonzero coefficient $a_{r,s}$ called the pivot element; r is called the pivot row and s the pivot column. The process of the selection of these indices will be discussed later.

Consider the system of equations:

$$\sum_{j=1}^{n} a_{i,j} x_j = b_i; 1 \leq i \leq m$$

Let $a_{r,s} \neq 0$. Then, pivot operation carried out on this element means the following changes to the system:

1. Replace E_r, the r^{th} equation by $\frac{1}{a_{r,s}} E_r$. Thus, $E_r^{new} = \frac{1}{a_{r,s}^{old}} E_r^{old}$.
2. $E_i^{new} = E_i^{old} - \frac{a_{i,s}^{old}}{a_{r,s}^{old}} E_r^{old} = E_i^{old} - a_{i,s}^{old} E_r^{new}; i \neq r$.

We can also view this operation in terms of the values of new numbers (after the pivot operation) in terms of the old ones as follows:

$$a_{r,j}^{new} = \frac{a_{r,j}^{old}}{a_{r,s}^{old}}; b_r^{new} = \frac{b_r^{old}}{a_{r,s}^{old}}$$

$$a_{i,j}^{new} = a_{i,j}^{old} - \frac{a_{i,s}^{old} a_{r,j}^{old}}{a_{r,s}^{old}}; b_i^{new} = b_i^{old} - \frac{a_{i,s}^{old}}{a_{r,s}^{old}} b_r^{old}; i \neq r$$

Finally, we can also see this operation in terms of matrices as follows:

$$[Ax = b] \Longleftrightarrow [E_{r,s}^{-1} Ax = E_{r,s}^{-1} b]$$

where $E_{r,s}$ is obtained by replacing the r^{th} column of an identity matrix by the s^{th} column of the matrix A. Such a matrix is called an *(column) elementary matrix.*

The important property of this operation is that the set of solutions is not altered by it. This is quite easy to see.

0.1.1 (Gauss-Jordan) Canonical Form:

If A in the above equations is of the form $A = [I\ \bar{A}]$, (possibly after permuting its columns), then we say that the equations are in *(Gauss-Jordan) canonical form.* Written in microscopic form this would look like;

$$\begin{bmatrix} x_1 & & & & & +\bar{a}_{1,m+1}x_{m+1} & \cdots & +\bar{a}_{1,n}x_n & = & \bar{b}_1 \\ & x_2 & & & & +\bar{a}_{2,m+1}x_{m+1} & \cdots & +\bar{a}_{2,n}x_n & = & \bar{b}_2 \\ & & \ddots & & & \vdots & \ddots & \vdots & \vdots & \vdots \\ & & & x_{m-1} & & +\bar{a}_{m-1,m+1}x_{m+1} & \cdots & +\bar{a}_{m-1,n}x_n & = & \bar{b}_{m-1} \\ & & & & x_m & +\bar{a}_{m,m+1}x_{m+1} & \cdots & +\bar{a}_{m,n}x_n & = & \bar{b}_m \end{bmatrix}$$

The first m variables are called *basic (dependent) variables* and the remaining are *nonbasic (independent) variables.*

Theorem 4 *Given a system $[Ax = b]$ and a set $\{x_1, x_2, ..., x_m\}$ of basic variables, there is at most one canonical form.*

Proof: Suppose that there are two canonical forms corresponding to the same system $[Ax = b]$ and the same set of basic variables. Let these be:

$$\begin{bmatrix} x_1 & & & & & +\bar{a}_{1,m+1}x_{m+1} & \cdots & +\bar{a}_{1,n}x_n & = & \bar{b}_1 \\ & x_2 & & & & +\bar{a}_{2,m+1}x_{m+1} & \cdots & +\bar{a}_{2,n}x_n & = & \bar{b}_2 \\ & & \ddots & & & \vdots & \ddots & \vdots & \vdots & \vdots \\ & & & x_{m-1} & & +\bar{a}_{m-1,m+1}x_{m+1} & \cdots & +\bar{a}_{m-1,n}x_n & = & \bar{b}_{m-1} \\ & & & & x_m & +\bar{a}_{m,m+1}x_{m+1} & \cdots & +\bar{a}_{m,n}x_n & = & \bar{b}_m \end{bmatrix}$$

and

$$\begin{bmatrix} x_1 & & & & & +\hat{a}_{1,m+1}x_{m+1} & \cdots & +\hat{a}_{1,n}x_n & = & \hat{b}_1 \\ & x_2 & & & & +\hat{a}_{2,m+1}x_{m+1} & \cdots & +\hat{a}_{2,n}x_n & = & \hat{b}_2 \\ & & \ddots & & & \vdots & \ddots & \vdots & \vdots & \vdots \\ & & & x_{m-1} & & +\hat{a}_{m-1,m+1}x_{m+1} & \cdots & +\hat{a}_{m-1,n}x_n & = & \hat{b}_{m-1} \\ & & & & x_m & +\hat{a}_{m,m+1}x_{m+1} & \cdots & +\hat{a}_{m,n}x_n & = & \hat{b}_m \end{bmatrix}$$

If the values of the nonbasic variables are fixed, then there is a unique set of values for the basic variables (this is the reason for calling the basic variables dependent). Since these two forms correspond to the same set of equations, the solution sets are the same. In particular, setting all nonbasic variables equal to zero, there is only one solution – $x_i = \bar{b}_i = \hat{b}_i; 1 \leq i \leq m$. This establishes that the right sides are equal. Now consider the solution obtained by setting all nonbasic variables except x_i equal to zero and $x_i = 1; m+1 \leq i \leq n$. The corresponding values of the basic variables are also unique – $x_j = \bar{b}_j - \bar{a}_{j,i} = \hat{b}_j - \hat{a}_{j,i}; 1 \leq j \leq m$. Since $\bar{b}_j = \hat{b}_j; 1 \leq j \leq m$, it follows that $\bar{a}_{j,i} = \hat{a}_{j,i}; 1 \leq j \leq m; m+1 \leq i \leq n$. Hence these two forms are identical. Hence the result.□

The same result can also be shown in matrix form. A pivot operation makes use of at most $O(n^2)$ operations (additions, subtractions, multiplications and divisions) for a matrix of size $n \times n$. It is the central operation in the simplex algorithm used to solve linear programs. We now describe this algorithm.

Consider the LP: $[\min \sum_{j=1}^{n} c_j x_j : \sum_{j=1}^{n} a_{i,j} x_j = b_i; 1 \leq i \leq m]$.

Assumptions:

1. The system is in canonical form and the right hand side numbers (r.h.s) are nonnegative. Such a canonical form is said to be a *feasible canonical form*.

2. The problem is nondegenerate.

3. $c_j = 0$ if the variable x_j is basic.

The solution obtained by setting the nonbasic variables equal to zero is called the "*current basic solution*". This solution has the basic variables equal to r.h.s. It is feasible if the canonical form is also feasible; i.e. the r.h.s is nonnegative.

If the values of all c_j are also nonnegative, then this solution is also optimal. If not, select a variable x_j with $c_j < 0$ and increase this variable until one (or more) of the basic variable goes down to zero. Please note that in this procedure, only one nonbasic variable is being increased from zero at any given time. We adjust the values of the basic variables so that the equations are satisfied at all times. If the nonbasic variable selected to be increased is x_s and the basic variable that drops to zero (assume for now only one does) is x_r. Now we need a new canonical form in which the roles of variables x_s and x_r are reversed; i.e. now x_s should be basic and x_r should be nonbasic. This is achieved by a pivot operation on $a_{r,s}$; the fact that x_r went down to zero when x_s was being increased assures us that $a_{r,s}$ is not only not zero, but positive. The new canonical form has as its current solution one which is "better" than the previous one. The process is repeated until we can not improve any more. At this point in time we know that c_j is nonnegative for all j. There is one other possibility – when we increase x_s none of the basic variable decreases. In this case it is clear that $a_{i,s} \leq 0 \; \forall \; i$. It should also be clear, that in this case, the objective function tends to $-\infty$ as we increase x_s. We denote this phenomenon as "Problem is unbounded". In practice, we have to check what we have missed. These are the only two methods of exit from the algorithm. Now we show that the algorithm does stop after a finite number of steps.

At any step of the algorithm, we have a canonical form that looks like:

$$\left[\begin{array}{cccccccccccc}
x_1 & & & & & +\bar{a}_{1,m+1}x_{m+1} & \cdots & +\bar{a}_{1,s}x_s & \cdots & +\bar{a}_{1,n}x_n & = & \bar{b}_1 \\
& x_2 & & & & +\bar{a}_{2,m+1}x_{m+1} & \cdots & +\bar{a}_{2,s}x_s & \cdots & +\bar{a}_{2,n}x_n & = & \bar{b}_2 \\
& & \ddots & & & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\
& & & x_{m-1} & & +\bar{a}_{m-1,m+1}x_{m+1} & \cdots & +\bar{a}_{m-1,s}x_s & \cdots & +\bar{a}_{m-1,n}x_n & = & \bar{b}_{m-1} \\
& & & & x_m & +\bar{a}_{m,m+1}x_{m+1} & \cdots & +\bar{a}_{m,s}x_s & \cdots & +\bar{a}_{m,n}x_n & = & \bar{b}_m \\
& & & & -z & +\vec{c}_{m+1}x_{m+1} & \cdots & +\bar{c}_s x_s & \cdots & +\bar{c}_n x_n & = & -\bar{z}^0
\end{array}\right]$$

Let us suppose that $\min_j \bar{c}_j = \bar{c}_s < 0$; hence the entering variable is x_s. Let us also suppose that the pivot operation at this step is around the element $\bar{a}_{r,s}x_s$; please note that in this case that $\bar{a}_{r,s} > 0; \bar{b}_r > 0$; and $\bar{c}_s < 0$. Hence

$$(-\bar{z}^0)^{new} = (-\bar{z}^0)^{old} - \frac{\bar{c}_s \bar{b}_r}{\bar{a}_{r,s}} \geq (-\bar{z}^0)^{old}$$

Strict inequality holds iff $\bar{b}_r > 0$ and this holds under nondegeneracy assumption. Thus, we can not have the same canonical form repeat. hence the same set of variables can not repeat as the basic set of variables. (Please note we are NOT making the assertion that a single variable can not reenter the basis; we are simply asserting that the WHOLE SET can not repeat). There are only finitely many ($\binom{n}{m}$) of such sets that are potential candidates for being basic. Hence the algorithm is finite under the nondegeneracy assumption. Please note that EVEN under no such assumption, the above inequality regarding the value of z is still true; srict inequality may not hold any longer. Thus, if the algorithm does repeat a set of basic variables, then in between such occurrences of the same basic set, the value of z does not change. Thus, if we have some other way of showing that a basic set does not repeat, we still have a valid finiteness proof. This is what is done for the degenerate case later.

0.1.2 Two Phases of the Simplex Method:

Recall that we needed a feasible canonical form to START the simplex algorithm. Now we take up the case when this is not readily available. Indeed, the problem might be infeasible in which case no such form exists. In this section, we will identify if the problem is infeasible and if it is feasible provide a feasible canoncial form. And all this is done using the simplex algorithm on an (related) artificial problem.

Consider the problem $[\min cx : Ax = b; x \geq 0]$. Without loss of generality, we can assume that $b \geq 0$. Now consider the following problem instead;

$$\begin{array}{c}
\min w \\
\sum_{j=1}^{n} a_{i,j}x_j + v_i = b_i \\
\sum_{j=1}^{n} c_j x_j - z = 0 \\
\sum_{i=1}^{m} v_i - w = 0 \\
x_j \geq 0; 1 \leq j \leq n \\
v_i \geq 0; 1 \leq i \leq m
\end{array}$$

This problem could also be written as: $[\min \sum_{i=1}^{m} v_i : Ax + Iv = b; x \geq 0; v \geq 0]$. If $\exists$ a feasible solution x^0, then $x^0 \geq 0; Ax^0 = b$. Now $(x^0, 0)$ is a feasible solution to the second problem and for this solution $w = 0$ and hence it is optimal for the second problem. Conversely, if the optimal value for the second problem is 0, then in this solution $v_i^* = 0$ for all i. Hence its x^* is a feasible solution for the original problem. It is easy to produce a feasible canonical form for this problem as shown below to start the simplex algorithm using :

$$\left[\begin{array}{cccccccccccc}
a_{1,1}x_1 & +a_{1,2}x_2 & & +a_{1,n}x_n & +v_1 & & & & & & = & b_1 \\
a_{2,1}x_1 & +a_{2,2}x_2 & & +a_{2,n}x_n & & +v_2 & & & & & = & b_2 \\
\vdots & \vdots & \ddots & \vdots & & & \ddots & & & & \vdots & \vdots \\
a_{m-1,1}x_1 & +a_{m-1,2}x_2 & & +a_{m-1,n}x_n & & & +v_{m-1} & & & & = & b_{m-1} \\
a_{m,1}x_1 & +a_{m,2}x_2 & & +a_{m,n}xn & & & & +v_m & & & = & b_m \\
c_1x_1 & +c_2x_2 & & +c_nx_n & & & & & -z & & = & 0 \\
 & & & & +v_1 & +v_2 & \cdots\ +v_{m-1} & +v_m & & -w & = & 0
\end{array}\right]$$

This can be converted to

$$\left[\begin{array}{cccccccccccc}
a_{1,1}x_1 & +a_{1,2}x_2 & & +a_{1,n}x_n & +v_1 & & & & & & = & b_1 \\
a_{2,1}x_1 & +a_{2,2}x_2 & & +a_{2,n}x_n & & +v_2 & & & & & = & b_2 \\
\vdots & \vdots & \ddots & \vdots & & & \ddots & & & & \vdots & \vdots \\
a_{m-1,1}x_1 & +a_{m-1,2}x_2 & & +a_{m-1,n}x_n & & & +v_{m-1} & & & & = & b_{m-1} \\
a_{m,1}x_1 & +a_{m,2}x_2 & & +a_{m,n}xn & & & & +v_m & & & = & b_m \\
c_1x_1 & +c_2x_2 & & +c_nx_n & & & & & -z & & = & 0 \\
-\sum_{i=1}^{m} a_{i,1}x_1 & -\sum_{i=1}^{m} a_{i,2}x_2 & & \sum_{i=1}^{m} a_{i,n}x_n & & & \cdots & & & -w & = & -\sum_{i=}^{m}
\end{array}\right.$$

by subtracting the first m rows from the last. Now we have a canonical form suitable to use the simplex algorithm on for the second problem. if at otimality, $w^* > 0$, then the original problem is infeasible. If $w^* = 0$, then we have the required canoncial form for the original problem to carry out the simplex algorithm on the original problem to optimality or unboundedness. Please note that the artificial problem (the second one) can not be unbounded. This completes the description of the two phases of the simplex method.