LINEAR PROGRAMMING AND EXTENSIONS

R. Chandrasekaran
The University of Texas at Dallas

August 28, 1998

Chapter 1

Linear Inequalities and Theorems of the Alternative

Reference: O.L. Mangasarian: Nonlinear programming

1.1 Existence Results:

Theorem 1 *Let A be a $m \times n$ matrix. The system:*

$$\begin{gathered} Ax \geq 0 \\ A^t y = 0; y \geq 0 \\ A_{1,.}x + y_1 > 0 \end{gathered}$$

has a solution. (Note: x is not required to be nonnegative in this).

Proof: By induction on m.

For $m = 1$;let $x = A_{1.}$; and $y_1 = \begin{cases} 1 & A_{1,.} = 0 \\ 0 & A_{1,.} \neq 0 \end{cases}$. Now assume that the result is true for $m \leq k$; and show it holds for $m = k + 1$. For this purpose let $A = \begin{matrix} \hat{A} \\ A_{k+1,.} \end{matrix}$. By induction hypothesis, we have $\hat{x}, \hat{y}$ satisfying:

$$\begin{gathered} A\hat{x} \geq 0 \\ \hat{A}^t \hat{y} = 0; \hat{y} \geq 0 \\ \hat{A}_{1,.}\hat{x} + \hat{y}_1 > 0 \end{gathered}$$

If $A_{k+1,.}\hat{x} \geq 0$, then take $x = \hat{x}; y = (\hat{y}, 0)$. If $A_{k+1,.}\hat{x} < 0$, then we apply induction hypothesis to the $k \times n$ matrix B:

$$B = \begin{bmatrix} B_{1,.} \\ . \\ . \\ B_{k,.} \end{bmatrix} = \begin{bmatrix} A_{1,.} + \lambda_1 A_{k+1,.} \\ . \\ . \\ A_{k,.} + \lambda_k A_{k+1,.} \end{bmatrix}$$

where $\lambda_j = -\frac{A_{j,.}\hat{x}}{A_{k+1,.}\hat{x}} \geq 0$.

$$\begin{gathered} Bw \geq 0 \\ B^t v = 0; v \geq 0 \\ B_{1,.}w + v_1 > 0 \end{gathered}$$

Let $x = w - \frac{A_{k+1,.}w}{A_{k+1,.}\hat{x}}\hat{x}$. then $\hat{A}x = Bw \geq 0; A_{k+1,.}x = 0$. Let $y = (v, \sum_{j=1}^{k} \lambda_j v_j); A^t y = B^t v = 0; y \geq 0$. $A_{1,.}x + y_1 = B_{1,.}w + v_1 > 0.\Box$

Corollary 2 *Let A be as before. Then the system:*

$$\begin{gathered} Ax \geq 0 \\ A^t y = 0; y \geq 0 \\ Ax + y > 0 \end{gathered}$$

has a solution.

Corollary 3 *Let A be a $m_1 \times n$ matrix and B a $m_2 \times n$ matrix. Then:*

$$\begin{gathered} Ax \geq 0; Bx = 0 \\ A^t y^1 + B^t y^2 = 0; y^1 \geq 0 \\ Ax + y^1 > 0 \end{gathered}$$

has a solution.

Proof: Apply the first corollary to $\begin{bmatrix} A \\ B \\ -B \end{bmatrix}$.

Corollary 4 *A $m_1 \times n; B$ $m_2 \times n; C$ $m_3 \times n; D$ $m_4 \times n$. Then the system:*

$$\begin{gathered} Ax \geq 0; Bx \geq 0; Cx \geq 0; Dx = 0 \\ A^t y^1 + B^t y^2 + C^t y^3 + D^t y^4 = 0; y^i \geq 0; 1 \leq i \leq 3 \\ Ax + y^1 > 0; Bx + y^2 > 0; Cx + y^3 > 0 \end{gathered}$$

has a solution.

Theorem 5 *(Slater):Either*

$$Ax > 0; Bx \geq 0; Bx \neq 0; Cx \geq 0; Dx = 0$$

has a solution or

$$\begin{gathered} A^t y^1 + B^t y^2 + C^t y^3 + D^t y^4 = 0 \\ [y^1 \geq 0; y^1 \neq 0; y^2 \geq 0; y^3 \geq 0] \text{ or} \\ [y^1 \geq 0; y^2 > 0; y^3 \geq 0] \end{gathered}$$

has a solution.

Theorem 6 *(Motzkin):Either*

$$Ax > 0; Cx \geq 0; Dx = 0$$

has a solution or

$$\begin{gathered} A^t y^1 + C^t y^3 + D^t y^4 = 0 \\ [y^1 \geq 0; y^1 \neq 0; y^3 \geq 0] \text{ or} \\ [y^1 \geq 0; y^3 \geq 0] \end{gathered}$$

has a solution.

Theorem 7 *(Tucker):Either*

$$Bx \geq 0; Bx \neq 0; Cx \geq 0; Dx = 0$$

has a solution or

$$\begin{gathered} B^t y^2 + C^t y^3 + D^t y^4 = 0 \\ y^2 > 0; y^3 \geq 0] \end{gathered}$$

Theorem 8 *Either*

$$\begin{gathered} [Ax \geq 0; Ax \neq 0; Bx \geq 0; ; Cx \geq 0; Dx = 0] \text{ or} \\ Ax \geq 0; Bx > 0; Cx \geq 0; Dx = 0 \end{gathered}$$

has a solution or

$$\begin{gathered} A^t y^1 + B^t y^2 + C^t y^3 + D^t y^4 = 0 \\ [y^1 > 0; y^2 \geq 0; y^2 \neq 0; y^3 \geq 0] \end{gathered}$$

has a solution.

Theorem 9 *(Gordon)Either*

$$Ax > 0$$

has a solution or

$$A^t y = 0; y \geq 0; y \neq 0$$

has a solution.

Theorem 10 *(Farkas)Either*

$$Ax \leq 0; bx > 0$$

has a solution or

$$A^t y = b; y \geq 0$$

has a solution.

Theorem 11 *(Stiemke): Either*

$$Bx \geq 0; Bx \neq 0$$

has a solution or

$$B^t y = 0; y > 0$$

has a solution.

Theorem 12 *(Gale): Either*

$$Ax = c$$

has a solution or

$$A^t y = 0; cy = 1$$

has a solution.

Theorem 13 *(Gale): Either*

$$Ax \leq c$$

has a solution or

$$A^t y = 0; cy = -1; y \geq 0$$

has a solution.

Chapter 2

Fourier Elimination

Jean Baptiste Joseph Fourier is considered by many to be the originator of linear programming. He gave two methods to solve linear programs: a geometric one that we now call the simplex method and an algebraic one that is called the Fourier elimination method. It is the second that we take up now. It has very powerful uses in proving theorems and is practically efficient in a very limited number of cases. Its average complexity is not known. To illustrate the method let us take a simple example:

$$\begin{gathered}\max 2x_1 + x_2 \\ x_1 + 2x_2 \leq 6 \\ x_1 + x_2 \geq 2 \\ x_1 - x_2 \geq 3 \\ x_1 \geq 0; x_2 \geq 0\end{gathered}$$

Rewriting this as a minimization problem and using z to indicate the function to be minimized we get the system of inequalities:

$$\begin{gathered}-2x_1 - x_2 \leq z \\ x_1 + 2x_2 \leq 6 \\ x_1 + x_2 \geq 2 \\ x_1 - x_2 \geq 3 \\ x_1 \geq 0; x_2 \geq 0\end{gathered}$$

This is the same as the system:

$$\begin{gathered}-(z/2) - (x_2)/2 \leq x_1 \\ 2 - x_2 \leq x_1 \\ 3 + x_2 \leq x_1 \\ 0 \leq x_1\end{gathered} \tag{2.1}$$

$$x_1 \leq 6 - 2x_2 \tag{2.2}$$

$$x_2 \geq 0 \tag{2.3}$$

We choose x_1 as the first variable to eliminate (any one except z will do). The set of inequalities in 2.1 give a lower bound on the variable x_1; those in 2.2 yield an upper bound and those in 2.3 do not involve this variable. We get a new inequality for each combination of an inequality in 2.1 with one in 2.2. Doing this we get the system:

$$\begin{aligned} -(z/2) - (x_2/2) &\leq 6 - 2x_2 \\ 2 - x_2 &\leq 6 - 2x_2 \\ 3 + x_2 &\leq 6 - 2x_2 \\ 0 &\leq 6 - 2x_2 \\ x_2 &\geq 0 \end{aligned}$$

Rearranging this we get:

$$\begin{aligned} x_2 &\leq 4 + (z/3) \\ x_2 &\leq 4 \\ x_2 &\leq 1 \\ x_2 &\leq 3 \\ 0 &\leq x_2 \end{aligned}$$

Now we eliminate x_2 and get the system (*We could remove the redundant inequalities before doing this and it will save us a lot of work later*):

$$\begin{aligned} 0 &\leq 4 + (z/3) \\ 0 &\leq 4 \\ 0 &\leq 1 \\ 0 &\leq 3 \end{aligned}$$

The last three of these are vacuous and can be thrown away. The first is the same as: $z \geq -12$. Hence the optimum value of z is -12 (since we want to minimize z). This requires x_2 to be equal to zero. This in turn means that x_1 must equal 6. Hence we have the optimal solution as well. This is the method.

The main drawback in the method is that to eliminate a variable from a system with m constraints and n variables the new system may have $O(m^2)$ constraints in one less variable. This tends to "*blow*" the system up in size. The one case where this does not happen is what is known as the *2-SAT problem in logic*. Here at each stage the only constraints are of the form:

$$\begin{aligned} x_i - x_j &\geq \alpha \\ x_j - x_i &\geq \beta \\ x_i + x_j &\geq \gamma \\ -x_i - x_j &\geq \delta \end{aligned}$$

Thus, their number is at most $4(n^2)$. *Note*: *If every constraint has at most two variables*, *then this condition is maintained throughout the process*. On the other hand, it may be possible to use this process in the reverse to reduce the size of the problem – we know of very little work in this direction.

The advantage of this method is that it can be used to prove theorems. We give below some examples:

Chapter 3