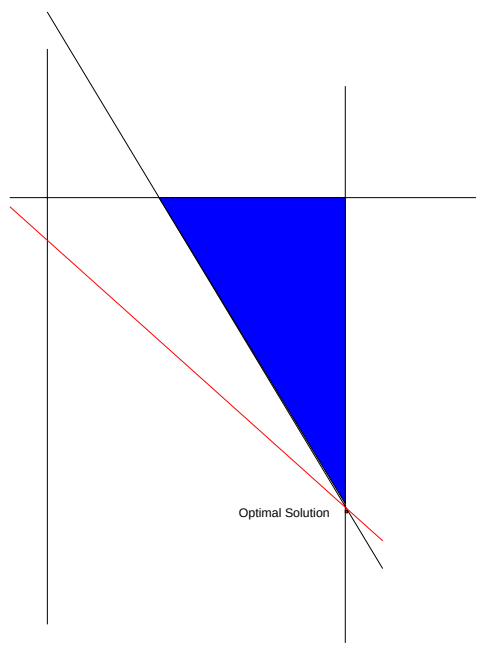
Lecture #13:

0.0.1 Linear Programming (Chapter 29)

Example 1

$$\begin{gathered}\min 40x_1 + 36x_2 \\ x_1 \leq 8 \\ x_2 \leq 10 \\ 5x_1 + 3x_2 \geq 45 \\ x_1 \geq 0; x_2 \geq 0\end{gathered}$$

Example 2 *When we have a problem in two variables, we can solve the problem graphically as shown below.*



When there are more than two variables (and there is no way you can see to reduce the number to two *without mutilating the problem*), then we cannot use the graphical method to solve the problem. For solving such problems, we have a method called the *simplex algorithm* that produces optimal solutions, indicates infeasibility or shows that the problem is unbounded, which ever is the case. Although the simplex algorithm is *theoretically* inefficient (in some sense), it works very well practically, and until recently, it was the most widely used algorithm. Now we are ready to describe the simplex algorithm to solve linear programs, and we begin by considering an "easy" example first to illustrate the logic. Consider the following example:

Example 3

$$\begin{gathered}\min 40x_1 + 36x_2 \\ x_1 \leq 8 \\ x_2 \leq 10 \\ 5x_1 + 3x_2 \leq 45 \\ x_1 \geq 0; x_2 \geq 0\end{gathered}$$

We now outline the steps involved in solving this problem:
Step 1: Convert each inequality to an equation by *introducing new variables called slack variables*. These newly introduced variables (as well as the old ones will be required to be nonnegative).

For example the first constraint becomes $x_1 + s_1 = 8$. Doing this to the entire problem results in:

$$\begin{gathered}\min 40x_1 + 36x_2 \\ x_1 + s_1 = 8 \\ x_2 + s_2 = 10 \\ 5x_1 + 3x_2 + s_3 = 45 \\ x_1 \geq 0; x_2 \geq 0; s_1 \geq 0; s_2 \geq 0; s_3 \geq 0\end{gathered}$$

Why can't we use the same slack more than once?
Step 2: Convert (if necessary) the problem to one of maximization (*this step is done solely to avoid confusion*). In our example, we want to minimize $40x_1 + 36x_2$ which is the same as maximizing $(-40x_1 - 36x_2)$; to make this into another equation, we let $z = -40x_1 - 36x_2$, or equivalently $z + 40x_1 + 36x_2 = 0$

Gathering all the equations and rewriting them in a form convenient for later use, we have:

$$\begin{bmatrix} z \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 0 & -40x_1 & -36x_2 \\ 8 & -x_1 & \\ 10 & & -x_2 \\ 45 & -5x_1 & -3x_2 \end{bmatrix}$$
$$x_i \geq 0; s_j \geq 0$$

Such a system of equations is called a **(Gauss-Jordan)-canonical form** of equations. The variables on the left side are called the **basic variables** *with respect to this canonical form* or, conversely, this is the canonical form with respect to this set of basic variables. The remaining variables are called **nonbasic variables**. The solution obtained by setting the nonbasic variables equal to zero is called the **current basic solution**. If it is nonnegative (except possibly in the z component), then it is also **feasible** and is called the **current basic feasible solution**. **The simplex algorithm requires a starting canonical form whose current basic solution is feasible**. We will show how this can always be achieved later by using an artificial problem from whose solution we will be able to find the solution of the original problem. Note that the first

equation implies $z = -(40x_1 + 36x_2) \leq 0$ for all values of x_i that are nonnegative. Our current solution has $z = 0$, and hence it is optimal; thus the optimal solution is $z = 0$; $s_1 = 8$; $s_2 = 10$; $s_3 = 45$; $x_1 = 0$; $x_2 = 0$. We were fortunate that we did not have to do any calculations in this example. In general, we will not be so fortunate, and we will now describe what to do in these cases by an illustration.

Example 4

$$\begin{gathered}\max 3x_1 + 2x_2 \\ -x_1 + 2x_2 \leq 4 \\ 3x_1 + 2x_2 \leq 14 \\ x_1 - x_2 \leq 13 \\ x_1 \geq 0; x_2 \geq 0\end{gathered}$$

After introducing slack variables and introducing the variable z, we get:

$$\begin{bmatrix} z \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 0 & 3x_1 & +2x_2 \\ 4 & x_1 & -2x_2 \\ 14 & -3x_1 & -2x_2 \\ 13 & -x_1 & +x_2 \end{bmatrix}$$
$$x_i \geq 0; s_j \geq 0$$

The current basic solution is $z = 0$; $s_1 = 4$; $s_2 = 14$; $s_3 = 13$; $x_1 = 0$; $x_2 = 0$. It is feasible *since all x_j and s_i are nonnegative.* This is equivalent to all numbers in the RHS being nonnegative except possibly the top one. It is not optimal since increasing either x_1 or x_2 will increase z (and hence improve) the solution. We *choose* to increase x_1 keeping all other nonbasic variables (which in this case is only x_2) at zero value. This corresponds to the one-at-a-time process which is the simplex algorithm. The more we increase this variable, the more the value of z, and hence the "better" is our solution. However, as we increase this variable, s_2 and s_3 decrease, and at some point, one of these reaches the value zero (while the other is still nonnegative), and we cannot increase this nonbasic variable any further. In our example, this value for x_1 is $\frac{14}{3}$. At this point, we have an improved solution:, $z = 14$, $s_1 = 4 + \frac{14}{3} = \frac{26}{3}$, $x_1 = \frac{14}{3}$, $s_3 = 13 - \frac{14}{3} = \frac{25}{3}$, $x_2 = 0$; $s_2 = 0$. *Please note that the variable that became zero (from a positive value) is s_2.* Now we would like a new canonical form in which the role of x_1 and s_2 are interchanged. For this purpose, we use the equation in which s_2 "appears" (it appears only once) and use this to "solve" for x_1 and substitute this expression in all other equations for x_1. Doing this results in the system:

$$\begin{bmatrix} z \\ s_1 \\ x_1 \\ s_3 \end{bmatrix} = \begin{bmatrix} 14 & -s_2 & \\ \frac{26}{3} & -\frac{1}{3}s_2 & -\frac{8}{3}x_2 \\ \frac{14}{3} & -\frac{1}{3}s_2 & -\frac{2}{3}x_2 \\ \frac{25}{3} & +\frac{1}{3}s_2 & -\frac{1}{3}x_2 \end{bmatrix}$$
$$x_i \geq 0; s_j \geq 0$$

At this point, we know that the current basic solution is optimal. *This is one kind of termination in which we find one or more optimal solutions.* Not all problems have this kind of termination. For example, consider:

$$\begin{gathered}\max 5x_1 + 3x_2 \\ -2x_1 + x_2 \leq 8 \\ -x_1 + 5x_2 \leq 10 \\ x_2 \leq 15 \\ x_1 \geq 0; x_2 \geq 0\end{gathered}$$

Using slack variables, this reduces to:

$$\begin{gathered}\begin{bmatrix} z \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 0 & +5x_1 & +3x_2 \\ 8 & +2x_1 & -x_2 \\ 10 & +x_1 & -5x_2 \\ 15 & & -x_2 \end{bmatrix} \\ x_i \geq 0; s_j \geq 0\end{gathered}$$

Increasing variable x_1 improves the solution, and *there is no limit to its increase since no basic variable decreases as it is increased. Hence, we can increase z without limit. Hence the problem is said to be* **unbounded**, *and there is no optimal solution in this case, and we terminate the algorithm.*

Now we consider another example to illustrate yet another feature.

Example 5

$$\begin{gathered}\max x_1 + 2x_2 + 3x_3 + 4x_4 \\ -2x_1 + 2x_2 + x_3 = 4 \\ 3x_1 + x_2 + x_4 = 6 \\ x_i \geq 0; i = 1, 2, 3, 4\end{gathered}$$

Although this is *not quite* in canonical form corresponding to a feasible basic solution, to render it so is quite easy. The given system is:

$$\begin{gathered}x_1 + 2x_2 + 3x_3 + 4x_4 = z \\ -2x_1 + 2x_2 + x_3 = 4 \\ 3x_1 + x_2 + x_4 = 6 \\ x_i \geq 0; i = 1, 2, 3, 4\end{gathered}$$

which can be rewritten as:

$$\begin{gathered}\begin{bmatrix} z \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & +x_1 & +2x_2 & +3x_3 & +4x_4 \\ 4 & +2x_1 & -2x_2 & & \\ 6 & -3x_1 & -x_2 & & \end{bmatrix} \\ x_i \geq 0; i = 1, 2, 3, 4\end{gathered}$$

Substituting for x_3 and x_4 in the z equation, we get the desired canonical form and can start the simplex algorithm. Such problems are said to be *in almost canonical form.*

Big-M Method:

Consider the example discussed at the outset:

Example I:

$$\begin{gathered}\min 40x_1 + 36x_2 \\ x_1 \leq 8 \\ x_2 \leq 10 \\ 5x_1 + 3x_2 \geq 45 \\ x_1 \geq 0; x_2 \geq 0\end{gathered}$$

This is almost like example II and setting up yields:

$$\begin{bmatrix} z \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 0 & -40x_1 & -36x_2 \\ 8 & -x_1 & \\ 10 & & -x_2 \\ -45 & +5x_1 & +3x_2 \end{bmatrix}$$
$$x_i \geq 0; s_j \geq 0$$

or:

$$\begin{bmatrix} z \\ s_1 \\ s_2 \\ -s_3 \end{bmatrix} = \begin{bmatrix} 0 & -40x_1 & -36x_2 \\ 8 & -x_1 & \\ 10 & & -x_2 \\ 45 & -5x_1 & -3x_2 \end{bmatrix}$$
$$x_i \geq 0; s_j \geq 0$$

Thus, we do not have a feasible canonical form. *We need* a basic variable in the last equation. *What we propose to do is to convert the problem to an almost canonical form that is feasible for a related but artificial problem*; *the solution of this artificial problem will lead us to one of two conclusions*: (i) *the original problem has an optimal canonical form, in which case one will be produced at optimality to the artificial problem, or* (ii) *the original problem does not have an optimal solution. In case* (ii) *we will, in some cases, be able to ascertain whether the original problem is feasible or not and, in some other cases, conclude that it* (*the original problem*) *is either infeasible or it is unbounded.* In either case, it does not have an optimal solution and we would stop. The related artificial problem for our example is:

Example I:

$$\begin{gathered}\min 40x_1 + 36x_2 + Mv_3 \\ x_1 \leq 8 \\ x_2 \leq 10 \\ 5x_1 + 3x_2 + v_3 \geq 45 \\ x_1 \geq 0; x_2 \geq 0; v_3 \geq 0\end{gathered}$$

Or Equivalently:

$$\begin{bmatrix} z \\ s_1 \\ s_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 & -40x_1 & -36x_2 & -M(45 - 5x_1 - 3x_2 + s_3) \\ 8 & -x_1 & & \\ 10 & & -x_2 & \\ 45 & -5x_1 & -3x_2 & +s_3 \end{bmatrix}$$
$$x_i \geq 0; s_j \geq 0; v_3 \geq 0$$

Here M is to be treated *as a big positive number*; *we never give it a specific value and treat it algebraically at all times.* We now have a feasible canonical form for the artificial problem and, therefore, can use the simplex algorithm for it. *Please make sure that the RHS will turn out to be nonnegative before introducing artificial variables.*

If there are feasible solutions to the original problem, then all optimal solutions to the artificial problem must have all artificial variables equal to zero. This is somewhat like saying, "You can take my car if you pay a billion dollars," instead of saying, "You cannot take my car."

While doing the algorithm, if an artificial variable v becomes nonbasic, you can discard that variable (please do not discard the original or slack variables which belong to the original problem). If all artificial variables are removed, then we have the original problem as before. In one case, we have reached optimality for the artificial problem. There are two subcases to consider: I(a): no artificial variable is in the basis at optimality implies that we have found an optimal solution to the original problem; I(b): there are artificial variables still in the basis (and hence still positive) implies that the original problem is infeasible. The other is unboundedness for the artificial problem. Again there are two subcases: II(a): no artificial variable in the basis implies that the original problem is unbounded; II(b): artificial variables still in the basis and hence positive implies that the original problem is *either unbounded or infeasible and we are unable to decide which is true given only this information.* This concludes the discussion of "Big-M method."

Unrestricted Variables:

Note that the algorithm assumes that all variables are required to be nonnegative. Consider the example:

Example V:

$$\begin{gathered}\max x_1 - 2x_2 + 3x_3 \\ x_1 + x_2 + x_3 \leq 7 \\ x_1 - x_2 + x_3 \geq 2 \\ 3x_1 - x_2 - 2x_3 = -5 \\ x_1 \geq 0; x_2 \geq 0; x_3 \text{ unrestricted in sign}\end{gathered}$$

There are two methods to resolve this; one is given in your book. We give another; both have merits. The selection of the method to use depends on the circumstance under consideration.

Using any equation (other than the objective function equation) that has a nonzero coefficient for the unrestricted variable, solve for that variable and eliminate it from all the remaining equations by substitution. The result is one less equation in one less variable. When the solution of this problem is found, back-substitute for the variable that was eliminated. *Why can we not do this with all variables*?

1 INTRODUCTION:

Problem 6 *The following problem is known as a linear programming Problem or simply as a 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 7 *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 8 *The most general form of a linear program has three kinds of constraints: $\{=, \leq, \geq\}$. 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:*

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

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 $\sin(\pi x_j) = 0$).

1.0.2 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 ith 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.

1.0.3 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 rth 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 rth column of an identity matrix by the sth 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.

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 9 *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:

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

and

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

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. This means that none of the rhs numbers are zero in any canonical form.

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:

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

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.

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

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

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.

2 Duality Theory

The pair of LPs :

$$\min cx : Ax \geq b; x \geq 0 \tag{1}$$

$$\max b^t y : A^t y \leq c^t; y \geq 0 \tag{2}$$

are said to be a *dual pair* of linear programs. One of these (arbitrarily chosen) is called the *primal* and the other its *dual*. It is easy to see that the dual of the dual is the primal. Since every linear program can be written in either of these forms we can restrict our analysis to these. Just for the sake of clarity we describe the relationship in greater detail.

If A is $m \times n$, c and x are $1 \times n$, b is $m \times 1$ then y is $m \times 1$. *Thus, there is a variable in the dual corresponding to every constraint in the primal and a constraint in the dual corresponding to every variable in the primal; what is the right hand side in the primal is the objective in the dual and conversely; minimization in the primal changes to maximization in the dual; the direction of the constraint inequalities changes.* This is best described in the following diagram:

Exercise:

Consider the general LP:

$$\begin{bmatrix} A^{1,1} & A^{1,2} & A^{1,3} \\ A^{2,1} & A^{2,2} & A^{2,3} \\ A^{3,1} & A^{3,2} & A^{3,3} \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \\ x^3 \end{bmatrix} \begin{matrix} \geq \\ \leq \\ = \end{matrix} \begin{bmatrix} b^1 \\ b^2 \\ b^3 \end{bmatrix}$$

$$x^1 \geq 0; x^2 \leq 0; x^3 \; unrestricted$$

$$\min[c^1x^1 + c^2x^2 + c^3x^3]$$

Transform it to the standard form and then write the dual; see if you can do the reverse of the transformation and set up the rules for writing dual in general.

2.0.1 Theorems Connecting the Two Problems

Theorem 10 *(Weak Duality): Let x^0 be any feasible solution to P and let y^0 be any feasible solution to D. Then $cx^0 \geq b^t y^0$.*

Corollary 11 *If x^0 and y^0 are as above and satisfy $cx^0 = b^t y^0$, then they are optimal to the respective problems.*

Proof: Since x^0 is feasible to P, $Ax^0 \geq b$; since y^0 is feasible to D, $y^0 \geq 0$. Hence

$$(y^0)^t Ax^0 \geq (y^0)^t b = b^t y^0$$

By using a similar argument we get

$$(x^0)^t A^t y^0 \leq (x^0)^t c^t = cx^0$$

Since

$$(y^0)^t Ax^0 = ((y^0)^t Ax^0)^t = (x^0)^t A^t y^0$$

weak duality theorem follows. The corollary is a simple consequence of the theorem and the definition of optimality.□

Theorem 12 *(Strong Duality): If both P and D are feasible then both have optimal solutions and $cx = b^t y$ for any pair of optimal solutions to the two problems.*

Proof: By weak duality theorem, neither problem is unbounded. Since $\exists$ a finite algorithm which terminates for any LP in one of three conditions: infeasibility, unboundedness, optimality and we have ruled out the first two the third must be true for both problems. Furthermore, the algorithm terminates with an optimal basis B^* which satisfies $\pi \geq 0$; $c - \pi A \geq 0$ where $\pi = c_{B^*}(B^*)^{-1}$. It is easy to see that $y^* = \pi^t$ is feasible to D. Since

$$b^t y^* = \pi b = c_{B^*}(B^*)^{-1} b = c_{B^*} x^*_{B^*} = cx^*$$

it follows from weak duality theorem, that y^* is optimal to D and hence the theorem.□

Theorem 13 *(Weak Complementary Slackness): Let x^* be any optimal solution to P and let y^* be any optimal solution to D. Then the following statements are true:*

$$\sum_j a_{ij} x^*_j > b_i \Longrightarrow y^*_i = 0; \sum_i a_{ij} y^*_i < c_j \Longrightarrow x^*_j = 0$$

$$\sum_j a_{ij}x_j^* = b_i \Longleftarrow y_i^* > 0; \sum_i a_{ij}y_i^* = c_j \Longleftarrow x_j^* > 0$$

Conversely if x^0 is a feasible solution to P and y^0 is a feasible solution to D satisfying:

$$\sum_j a_{ij}x_j^0 > b_i \Longrightarrow y_i^0 = 0; \sum_i a_{ij}y_i^0 < c_j \Longrightarrow x_j^0 = 0$$

$$\sum_j a_{ij}x_j^0 = b_i \Longleftarrow y_i^0 > 0; \sum_i a_{ij}y_i^0 = c_j \Longleftarrow x_j^0 > 0$$

then these are optimal to the respective problems.

Proof: By strong duality theorem, $cx^* = b^t y^*$. Since x^* and y^* are feasible, by weak duality theorem,

$$cx^* = (x^*)^t c^t \geq (x^*)^t A^t y^* = (y^*)^t A x^* \geq (y^*)^t b = b^t y^*$$

Since the first and the last terms are equal everything in between is also equal to these quantities. Hence, $(x^*)^t(c^t - A^t y^*) = 0$; but $x^* \geq 0$ and $c^t - A^t y^* \geq 0$. Thus, the two conditions:

$$\sum_i a_{ij}y_i^* < c_j \Longrightarrow x_j^* = 0; \sum_i a_{ij}y_i^* = c_j \Longleftarrow x_j^* > 0$$

follow. The other two conditions are shown by a similar argument. To show the converse, we use weak duality theorem and the fact that these conditions imply that $cx^0 = b^t y^0$ and then use the corollary.□

3 Algorithms Based on Duality

There are several algorithms based on these theorems. First is the set of algorithms that are commonly known as variants of the simplex method. This class includes: (i)the dual simplex method; and (ii)the self dual algorithm in addition to the *regular* simplex now called the primal simplex method. These are described now:

3.0.2 Dual Simplex Algorithm:

Here to we start with a canonical form; but instead of requiring the $\bar{b}$ to be nonnegative as in the regular simplex, we need the vector $\bar{c}$ to be nonnegative. Just as the regular simplex maintains the nonnegativity of $\bar{b}$, the dual simplex maintains $\bar{c}$ nonnegative. Just as the primal simplex stops with optimality when $\bar{c}$ becomes nonnegative, the dual simplex stops with optimality when $\bar{b}$ becomes nonnegative. For this reason, the condition that $\bar{b} \geq 0$ is often referred to as primal feasibility where as the condition of $\bar{c} \geq 0$ is referred to as dual feasibility.

These also correspond to bases that provide feasible solutions to the primal and the dual problems respectively.

There is the case of unboundedness in the primal algorithm as one of the terminating conditions. this is the same as dual infeasibility. This occurs when we have a column with $\bar{c}_j < 0$;and $\bar{a}_{i,j} \leq 0 \; \forall i$. The corresponding case in the dual simplex is termination with indication of primal infeasibility which occurs when there is a row with $b_i < 0$; and $a_{i,j} \geq 0 \; \forall \; j$.

In the primal algorithm we select a column with $\bar{c}_j < 0$ if one exists. In the dual algorithm we select a row with $\bar{b}_i < 0$ if one exists. In the primal algorithm we select a row so that new $\bar{b}$ will remain nonnegative; in the dual, we select a column so that the new $\bar{c}$ will remain nonnegative.

Finiteness in the primal rests on the fact that the value of z is decreasing; the corresponding argument for the dual algorithm is that it is increasing; keep in mind that at any step we do not have feasibility for the primal in the dual algorithm.

The primary reasons for the need for such an algorithm occurs in two areas. The first is sensitivity analysis when we change the rhs and the new value is out of the range that has the current basis feasible. it is certainly dual feasible since the primal algorithm stops with optimality only when $\bar{c} \geq 0$. In this case we can continue if we use the dual algorithm. Another application is when we have to add "forgotten" constraints and the old solution does not satisfy the new constraints. This especially occurs when we try to solve integer programs using LP.

3.1 Self Dual Algorithm

Here we start with any canonical form. Let $\bar{b}_i^{mod} = \bar{b}_i + \theta$ whenever $\bar{b}_i < 0$ and also let $\bar{c}_j^{mod} = \bar{c}_j + \theta$ whenever $\bar{c}_j < 0$. For θ sufficiently large, this basis will be optimal. As θ decreases, one of these values goes to zero first; if it is one of the rhs numbers then follow the primal simplex; else follow the dual simplex. Finiteness follows from the fact that θ keeps decreasing until it either goes to zero or there is an indication of either primal or dual infeasibility.

This completes the description of variants of the simplex that we will discuss. There is one other variant due to Wang that we will not discuss here. We will now consider algorithms based on complementary slackness theorems. These preserve complementary slackness but not primal or dual feasibility nor do they rely on bases. These are therefore, not simplex type algorithms in general. The foremost among these is the primal-dual algorithm which we now take up.

4 Primal Dual Algorithm

Consider the pair P & D of LPs :

$$\min cx : Ax = b; x \geq 0 \tag{3}$$

$$\max b^t y : A^t y \leq c^t \tag{4}$$

Primal dual algorithm starts with (an easily found) feasible (but not necessarily basic) solution y^0 to D. Let $S = (j : \sum_i a_{ij} y_i^0 = c_j)$. *If y^0 were an optimal solution to D, then complementary slackness would require $x_j = 0 \ \forall \ j \notin S$ in any optimal solution to P and, conversely, any such feasible solution to P is optimal. The primal dual algorithm behaves as if y^0 is optimal to D and hence tries to find a feasible solution to P with the additional condition that* $x_j = 0 \, \forall \, j \notin S$. We try to find

$$x^S : x^S \geq 0; A^S x^S = b \tag{5}$$

This is called *the restricted primal problem.* If we succeed, then the pair (x^0, y^0) is optimal to P and D, respectively (with $x_j^0 = x_j^S$ for $j \in S$ and $x_j^0 = 0$ for $j \notin S$); this is because they are feasible and satisfy complementary slackness conditions. If not, we have the optimal dual solution σ^* to the phase I problem for this restricted primal problem:

$$\min \sum_i v_i : A^S x^S + Iv = b; x^S \geq 0; v \geq 0$$

which is used to produce a "*better*" solution to D, and the entire process is repeated. The algorithm terminates with one of two conclusions: (i) exhibiting an optimal pair of solutions, or (ii) indicating infeasibility of P and hence unboundedness of D.

4.1 Details

1 If y is a feasible solution to D and x is feasible to P in which $\sum_i a_{ij} y_i < c_j \Longrightarrow x_j = 0$, then these are optimal to P and D, respectively.

2 Let y^0 be feasible to D. Let S be defined as above. Let σ^* be an optimal solution to the dual of the LP:

$$\min \sum_i v_i : A^S x^S + Iv = b; x^S \geq 0; v \geq 0$$

and suppose that the optimal value of this LP is positive. Thus, we cannot find $x^S \geq 0$ satisfying $A^S x^S = b$. σ^* satisfies the relations:

$$(A^S)^t \sigma^* \leq 0; \sigma^* \leq e$$
$$w^* = b^t \sigma^* > 0$$

(a)By Farkas Lemma, we have:

$A^t \sigma^* \leq 0 \Longrightarrow \exists$ no solution to: $x \geq 0; Ax = b$ since $b^t \sigma^* > 0$.

(b)$[A^t\sigma^* \leq 0]$ is false $\Longrightarrow \exists\ \theta > 0 \ni y^1 = y^0 + \theta\ \sigma^*$ is feasible to D and satisfies the relations: $b^t y^1 > b^t y^0$ and $\sum_i a_{ij} y_i^1 = c_j$ for some $j \notin S$.

Proof:(b)$\sum_i a_{ij} y_i^1 = \sum_i a_{ij} y_i^0 + \theta \sum_i a_{ij}\sigma_i^*$. For $j \in S$, $\sum_i a_{ij} y_i^0 = c_j$ and $\sum_i a_{ij}\sigma_i^* \leq 0$ and hence $\sum_i a_{ij} y_i^1 \leq c_j$ for $\forall\ \theta > 0$. For $j \notin S$, since $\sum a_{ij} y_i^0 < c_j$, $\exists\ \theta > 0 \ni \sum a_{ij} y_i^1 \leq c_j$, and if we choose θ as large as possible without violating this condition, equality will hold for at least one $j \notin S$ as promised. (Note: If for some $j \in S$, x_j is in the optimal basis in the restricted primal, then for this j, $\sum_i a_{ij}\sigma_i^* = 0$ and hence $\sum a_{ij} y_i^1 = c_j$ for this j; hence we "*lose*" only those variables that are not in the optimal basis of the restricted primal.) This means that we can continue the restricted primal from where we left off.

$$b^t y^1 = b^t y^0 + \theta b^t \sigma^* > b^t y^0$$

since $\theta > 0$ and $b^t\sigma^* = w^* > 0$.

3 If we used the "*lexicographic*" version of the simplex method for the restricted primal, then the whole algorithm is finite.

Proof:Since the objective of the restricted primal decreases lexicographically, not only is it finite but we cannot return to the same restricted problem.

4 $c_j - \sum_i a_{ij} y_i^1 = \bar{c}_j^{new} +_{\theta} a_{ij} - \theta \sum_i a_{ij}\sigma_i^*$

(Note: $\bar{d}_j = 0 - \sum_i a_{ij}\sigma_i^* = -\sum_i a_{ij}\sigma_i^*\ \forall\ j$). This provides an easy way to produce the new S and θ^* the largest value of $\theta \ni y^1$ is feasible to D.

5 *Note*: *It is not necessary that the restricted primal be solved using some form simplex method. Indeed, in most applications of the primal dual algorithm this is not done. A special algorithm is used that not only solves the restricted primal but also provides σ^* that is needed to implement primal dual algorithm. We will use the transportation problem as an illustration in this course. Matching will be another example solved using this algorithm in the networks course.*

Now we take up the special case of the transportation and assignment problems for which the primal dual algorithm is especially suited. Indeed, these were the first problems on which this algorithm was used; the high degree of degeneracy in these problems made the regular simplex unsuitable for these problems.

5 Transportation and Assignment Problems

In an introductory course in operations research one of the first topics introduced is the transportation problem. Its history dates back at least to the works of

L.Kantarovitch and **F.L. Hitchcock**. The problem stated as a linear program is:

$$\begin{array}{c} [\min \sum_i \sum_j c_{i,j} x_{i,j} \\ \sum_j x_{i,j} = a_i \\ \sum_i x_{i,j} = b_j \\ x_{i,j} \geq 0; 1 \leq i \leq m; 1 \leq j \leq n \end{array} \tag{6}$$

Its LP dual is

$$\begin{array}{c} \max \sum_i a_i u_i + \sum_j b_j v_j \\ u_i + v_j \leq c_{i,j}; 1 \leq i \leq m; 1 \leq j \leq n] \end{array} \tag{7}$$

We illustrate the *primal dual algorithm of linear programming* on this problem. We will use such an algorithm in matching and other problems later on. This type of algorithm starts with an (easily found) feasible (*not necessarily basic*) solution to the dual. In our example, start with any set of values for $\{u_i\}$. Let $v_j = \min_i [c_{i,j} - u_i]$. The set of values usually chosen for u_i is $u_i^0 = \min_j c_{i,j}$ and this yields $v_j^0 = \min_i [c_{i,j} - u_i^0]$.

Now the algorithm forces the values of $x_{i,j}$ for which $u_i + v_j < c_{i,j}$ to zero. Using only the remaining variables we *try* to find a feasible solution to the primal problem; if we succeed, then any such solution is optimal to the original problem. If we fail to find such a solution, then we modify the dual solution to a *better* one and repeat the process. *The problem of trying to find a feasible solution to the primal under the restriction that some variables be zero is called the restricted primal problem.* We will solve the restricted primal problem by solving a maximal flow problem which is described next before we resume the discussion of the primal dual algorithm.

For this purpose we construct a restricted network: $G^r = [N = \{s\} \cup \{t\} \cup S \cup T; A^r = \{(i,j) : u_i + v_j = c_{i,j}\} \cup \{(s,i)\} \cup \{(j,t)\}]$. Such a network is sometimes called the *equality graph* for obvious reasons. s is the origin and t is the destination and the capacity of arc (s,i) is a_i and that of (j,t) is b_j. Capacity of arcs of type (i,j) with $i \in N$ and $j \in N$ is ∞. Since $\sum_i a_i = \sum_j b_j$, if maximum flow in the above network is equal to this amount, then and only then we have a feasible flow to the original problem; and in this case this flow is optimal as claimed before by duality theory of LP. If max flow is less than this amount, then we get a minimal cut separating s and t and we use this cut to improve the dual solution. Such a minimal cut splits the set S, of supply points as well as the set T, of demand points. Let the labeled nodes be $I \subset S$ and $J \subset T$; and let the set of unlabeled nodes be $\bar{I} \subset S$ and $\bar{J} \subset T$. As mentioned before if $I = J = \phi$, then we are done. If not let $\delta = \min_{i \in I; j \in \bar{J}} [c_{i,j} - u_i - v_j] > 0$. The last statement is true since we can not have an arc of the form $i \in I$, and $j \in \bar{J}$ in the restricted network. Now change the dual solution as follows:

$$u_i' = \begin{cases} u_i & i \notin I \\ u_i + \delta & i \in I \end{cases}$$
$$v_j' = \begin{cases} v_j & j \notin J \\ v_j + \delta & j \in J \end{cases}$$

Note that $c_{i,j} - u'_i - v'_j = c_{i,j} - u_i - v_j$ for (i,j) of either of two types:(i)$i \in I$ and $j \in J$ or (ii)$i \notin I$ and $j \notin J$. For the case $i \notin I$ and $j \in J$, $c_{i,j} - u'_i - v'_j = c_{i,j} - u_i - v_j + \delta \geq 0$. For the case $i \in I$ and $j \notin J$, the choice of δ is such that $c_{i,j} - u'_i - v'_j \geq 0$. Hence the new u_i and v_j form a feasible solution to the dual.

$$\sum_i a_i u'_i + \sum_j b_j v'_j = \sum_i a_i u_i + \sum_j b_j v_j + \delta[\sum_{i \in I} a_i - \sum_{j \in J} b_j].$$

Since we know that the current flow F across the network satisfies:

$$F = [\sum_{i \notin I} a_i + \sum_{j \in J} b_j] < \sum_{i \in N} a_i$$

it follows that:

$$\sum_{i \in I} a_i - \sum_{j \in J} b_j > 0$$

Hence the new dual solution is an improved solution as claimed. The new equality graph will have all old arcs of the type (i) or (ii) above. It will have at least one new arc of the type $i \in I$ and $j \notin J$. *We may lose some arcs of the type* $i \notin I$ and $j \in J$; but all these arcs have zero flow at the point in time we made a dual variable change. *This allows us to retain the old flows and proceed further without redoing all this work from scratch*; *indeed*, *we may even retain the old labels in flow labeling and increase the set of labeled nodes because of the new arc of type* $i \in I$ and $j \notin J$. Thus, each time we do a dual variable change the set of labeled nodes enlarges and hence we can not have a continuous string of dual variable changes for more than $|S \cup T|$ number of iterations; at the end of this set of dual variable changes total flow must increase. If all data are integral this alone guarantees finiteness of the algorithm. If the data are not integral, then we need to use a finite algorithm for flow maximization at each step. If we do this finiteness is guaranteed because flow will increase in at most $|S \cup T|$ steps and each time we do a dual variable change, the current flow equals $[\sum_{i \notin I} a_i + \sum_{j \in J} b_j]$ and therefore the same I, J combination can not repeat. The number of such combinations is finite and hence the algorithm is also finite. *This last argument is very useful to show finiteness of algorithms.* In order to make the algorithm polynomial, we need to use scaling techniques as in **Edmonds and Karp.**

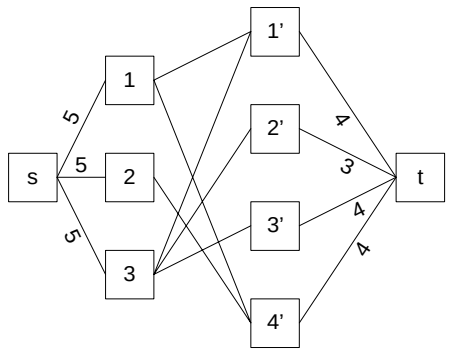
Example 14

$\frac{j\rightarrow}{i\downarrow}$	1'	2'	3'	4'	a_i
1	2	2	2	1	5
2	10	8	5	4	5
3	7	6	6	8	5
b_j	4	3	4	4	

The data in the last row and column are supply and demand quantities; the reamining numbers are the unit costs.

Step 1. The starting dual solution is: $u^0 = (1, 4, 6)^t; v^0 = (1, 0, 0, 0)^t$.

$A^0 = \{(1, 1'); (1, 4'); (2, 4'); (3, 1'); (3, 2'); (3, 3')\}$. G^0 is shown below:

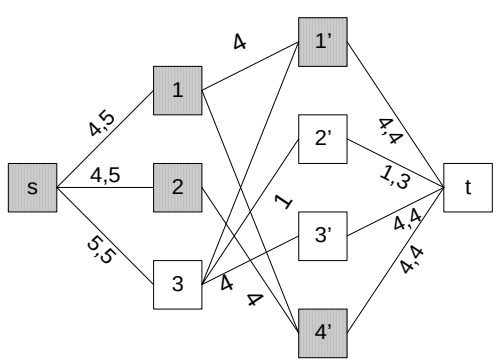


$I = \{1, 2\}; J = \{1', 4'\}; \bar{I} = \{3\}; \bar{J} = \{2', 3'\}; \delta = \min[1, 1, 4, 1] = 1$

The new solution is given by:

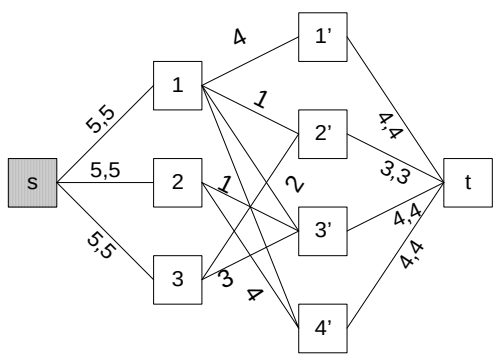
$$\left[\begin{array}{cccccc} \frac{v\rightarrow}{u\downarrow} & 0 & 0 & 0 & -1 & \\ 2 & 4 & & & & 5 \\ 5 & & & & 4 & 5 \\ 6 & & 1 & 4 & & 5 \\ & & & & & \\ & 4 & 3 & 4 & 4 & \frac{a\uparrow}{\leftarrow b} \end{array}\right]$$

$A^1 = \{(1, 1'); ((1, 2'); (1, 3'); (1, 4'); (2, 3'); (2, 4'); (3, 2'); (3, 3')\}$. G^1 is shown below:



The new solution is:

$$
\begin{bmatrix}
\frac{v\rightarrow}{u\downarrow} & 0 & 0 & 0 & -1 & \\
2 & 4 & 1 & & & 5 \\
5 & & & 1 & 4 & 5 \\
6 & & 2 & 3 & & 5 \\
 & & & & & \\
 & 4 & 3 & 4 & 4 & \frac{a\uparrow}{\leftarrow b}
\end{bmatrix}
$$



Since this is feasible to the primal, it is optimal by complementary slackenss therroem.