# LINEAR PROGRAMMING AND EXTENSIONS

R. Chandrasekaran
UT Dallas

August 28, 1998

# Chapter 1

# Duality Theory

The pair of LPs :

$$\min cx : Ax \geq b; x \geq 0 \tag{1.1}$$

$$\max b^t y : A^t y \leq c^t; y \geq 0 \tag{1.2}$$

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:

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

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

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

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.

## 1.1 Theorems Connecting the Two Problems

**Theorem 1** *(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 2** *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 3** *(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 ∃ 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 4** *(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,

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

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.□

*Please note that all these proofs are made possible by the existence of an algorithm. Such proofs are called algorithmic proofs.* We will now give a purely algebraic proof of these results. There is a set of results that together go under the name "*Theorems of the Alternative*". Each of these is of the form: "Exactly one of a given set of alternatives is true". In many of these there are two alternatives labeled I and II. The following is a preliminary set that we need now (we will mention a few more and show how these are proved later on).

**Theorem 5**

$$\exists\, x \ni Ax = b \tag{1.3}$$

$$\exists\, y \ni yA = 0, yb \neq 0 \tag{1.4}$$

**Theorem 6**

$$\exists\, x \ni Ax = b, x \geq 0 \tag{1.5}$$

$$\exists\, y \ni yA \geq 0; yb < 0 \tag{1.6}$$

**Theorem 7**

$$\exists\, x \ni Ax \geq b; x \geq 0 \tag{1.7}$$

$$\exists\, y \ni yA \leq 0; yb > 0 \tag{1.8}$$

Theorem A deals with solvability of equations and *the inconsistent equation*. Theorem B is called the *Farkas Lemma* and deals with solvability of inequalities. Theorem C is an alternative form of B.

**Definition 1** *A square matrix $K$ is said to be* skew (anti) symmetric *if $K^t = -K$ (i.e. $k_{ji} = -k_{ij} \forall\, i, j$).*

**Theorem 8** *If $K$ is a skewsymmetric matrix then the system::*

$$x \geq 0, Kx \geq 0, Kx + x > 0 \tag{1.9}$$

*always has a solution.*

**Proof:** First we will show that the system :

$$x^i \geq 0, Kx^i \geq 0, (Kx^i + x^i)_i > 0 \tag{1.10}$$

has a solution. Then letting $x = \sum_i x^i$ proves the theorem. For this purpose let $b = e_i$ (the $i^{th}$ unit vector), and $A = K$ in theorem C. If C.I is true, then

$$\exists\, x^i \ni Kx^i \geq e_i; x^i \geq 0$$

This is the required $x^i$. If C.II is true in $C$ then

$$\exists\, y \ni Ky = -K^t y \geq 0; y \geq 0; e_i^t y = y_i > 0$$

Let $x^i = -y$. Thus, in any case, we have the theorem.□

In theorem D, if we let

$$K = \begin{bmatrix} 0 & A & -b \\ -A^t & 0 & c^t \\ b^t & -c & 0 \end{bmatrix}$$

note that $K$ is $(m+n+1)$x$(m+n+1)$ then we get the results needed for duality. We indicate these below.

The immediate consequences of theorem $D$ are: $\exists$ vectors $x^0$, $y^0$, and a scalar $t^0$ satisfying the relations:

$$Ax^0 - t^0 b \geq 0; y^0 \geq 0; Ax^0 - t^0 b + y^0 > 0$$
$$-A^t y^0 + t^0 c^t \geq 0; x^0 \geq 0; -A^t y^0 + t^0 c^t + x^0 > 0$$
$$b^t y^0 - cx^0 \geq 0; t^0 \geq 0; b^t y^0 - cx^0 + t^0 > 0$$

**Theorem 9** *Exactly one of the following alternatives is true:*

I Both $P$ and $D$ have optimal solutions and their values are equal.

II One of the problems is infeasible and the other is feasible but unbounded.

III Neither problem is feasible.

**Proof**:We know that $x^0, y^0$, and $t^0$ of the type mentioned above exist. There are two cases to consider:

Case (i) $t^0 > 0$: Let $(x^*, y^*) = (x^0, y^0)/t^0$. It is easy to verify that these are optimal to $P$ and $D$ respectively.

Case(ii) $t^0 = 0$: Then we have

$$Ax^0 \geq 0; y^0 \geq 0; A^t y^0 \leq 0; x^0 \geq 0; b^t y^0 > cx^0$$

If one of the two problems is feasible (say $P$), then $\exists\, x^1 \ni Ax^1 \geq b; x^1 \geq 0$. From these it follows that $(y^0)^t Ax^1 \geq (y^0)^t b = b^t y^0; b^t y^0 \leq 0$ since $A^t y^0 \leq 0, x^1 \geq 0$, and $(y^0)^t Ax^1 = (x^1)^t Ay^0$. Hence $cx^0 < 0$ and this combined with $Ax^0 \geq 0 \Longrightarrow P$ is unbounded. If $D$ is the one that is feasible then its unboundedness is shown similarly. This completes the proof of the theorem; the equality of the values is shown by combining these with weak duality theorem shown before. □

Weak complementary slackness is shown as before (recall that the proofs were algebraic). We instead show :

**Theorem 10** *(Strong) Complementary Slackness: If both problems have optimal solutions, then $\exists$ a pair $x^*, y^*$ of optimal solutions to the two problems* $\ni$

$$\sum_j a_{ij} x_j^* > \Longleftrightarrow y_i^* = 0 \;\forall\, i; \sum_i a_{ij} y_i^* < c_j \Longleftrightarrow x_j^* = 0 \;\forall\, j$$

**Proof**: We have the case (i) above; $x^*, y^*$ defined there will do the job.□

## 1.2 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:

#### 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.

### 1.2.1 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 $\theta$ sufficiently large, this basis will be optimal. As $\theta$ 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 $\theta$ 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.

## 1.3 Primal Dual Algorithm

Consider the pair $P$ & $D$ of LPs :

$$\min cx : Ax = b; x \geq 0 \tag{1.11}$$

$$\max b^t y : A^t y \le c^t \tag{1.12}$$

*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

$$x^S : x^S \ge 0; A^S x^S = b \tag{1.13}$$

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 $\sigma^*$ to the phase I problem for this restricted primal problem:

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

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 $D$.

### 1.3.1 Details

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

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

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

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

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

(a)By Farkas Lemma, we have:

$A^t \sigma^* \le 0 \implies \exists$ no solution to: $x \ge 0; Ax = b$ 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 $\theta$ 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 $c_j - \sum_i a_{ij} y_i^1 = \bar{c}_j^{new} + \underset{\theta}{a_{ij}} - \theta \sum_i a_{ij}\sigma_i^*$

(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 $\theta^*$ 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 $\sigma^*$ 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.*

### 1.3.2 Open Question

Consider the following implementation of primal dual algorithm: Start as usual with $y^0$ feasible to $D$. Let there be a *black box* that has as input $y^0$ and as output either an $x^s \ni x^s \geq 0, A^s x^s = b$ or the set $(\sigma^*, ((x^s)^*, v^*))$ required as in the above description. Would the entire algorithm be finite ?

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.

## 1.4 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:

$$\begin{array}{c} [\min \sum_i \sum_j c_{i,j} x_{i,j} \\ \sum_j x_{i,j} = a_i \\ \sum_i x_{i,j} = b_j \\ x_{i,j} \geq 0; 1 \leq i \leq m; 1 \leq j \leq n \end{array} \tag{1.14}$$

Its LP dual is

$$\begin{array}{c} \max \sum_i a_i u_i + \sum_j b_j v_j \\ u_i + v_j \leq c_{i,j}; 1 \leq i \leq m; 1 \leq j \leq n] \end{array} \tag{1.15}$$

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.

## 1.5 Single Commodity Maximum Flows

In this section, we consider single commodity flow problems. The first of these is called the *maximum flow* problem and its description is given below:

**PROBLEM:** Given a directed network $G = [N; A]$, a special node, $s$, called the *source* or *origin* and a node, $t$, called the *sink or destination* and positive numbers $u_{i,j}$ representing the capacity of arc $(i, j) \in A$ we want to maximize the total shipment from $s$ to $t$. This problem is called the **maximum flow problem** in the literature and is one starting point in this area.

### 1.5.1 FORMULATION

Let $f_{i,j}$ be the flow on arc $(i,j) \in A$ and $F$ be the total flow across the network. Then:

$$\sum_i (f_{i,j} - f_{j,i}) = \begin{cases} F & i = s \\ 0 & i \neq s,t \\ -F & i = t \end{cases} \tag{1.16}$$

$$0 \leq f_{i,j} \leq u_{i,j} \; \forall \; (i,j) \in A$$

$$\max F$$

This is, of course, a linear program and can, in principle, be solved by the simplex method. However, the problem could be highly degenerate and unless special precautions are taken, it could be very inefficient to use the simplex method. Before describing methods to solve the problem we remark that it is easy to find a feasible solution for this problem as it has been stated. For example $[F = 0; f_{i,j} = 0 \; \forall \; (i,j) \in A]$ will suffice. First some definitions and a few preliminary results.

The notion of a *cut or a cut set separating* $s$ and $t$ can be defined in several slightly different ways. The first is to think of it as a partition of the node set $N$ into two disjoint sets $S$ and $\bar{S}$ with $s \in S$ and $t \in \bar{S}$. The next is to think of a cut as the set of arcs that connect these two sets mentioned above. A slightly weaker version is a minimal set of arcs whose removal disconnects the nodes $s$ and $t$. Unless otherwise stated, we will use the first form.

**Lemma 11** *Let $[f,F]$ be any feasible flow and let $(S,\bar{S})$ be any cut separating $s$ and $t$. Then $F \leq u(S,\bar{S})$ where $u(S,\bar{S}) = \sum_{i\in S; j\in\bar{S}} u_{i,j}$. $u(S,\bar{S})$ is called the* value *of the cut $(S,\bar{S})$. If equality holds then $[F,f]$ is optimal to the maximum flow problem and $(S,\bar{S})$ is a cut whose value is minimum among all cuts separating $s$ and $t$.*

**Proof**: The first part of the lemma clearly implies the second. To prove the first consider 1.16. Adding the equations corresponding to nodes in $S$ we get:

$F = \sum_{i\in S}\sum_{j\in N}(f_{i,j} - f_{j,i}) = f(S,N) - f(N,S)$
$= f(S,\bar{S}) - f(\bar{S},S) \leq u(S,\bar{S})$.

The last of these inequalities follows from the fact that $0 \leq f_{i,j} \leq u_{i,j}$ for all arcs $(i,j)$ in the network.□

The following algorithm, known in the literature as the *(flow) labeling algorithm,* is one way to show that there exist solutions that achieve equality in the relation above. It is usually attributed to **Ford** and **Fulkerson**.

### 1.5.2 Labeling Algorithm

**Input**: A feasible solution $[f,F]$.
**Step 0**: Label $s$ (-, $\infty$) and let $s \in S$, the set of labeled nodes.

**Step 1**: If $t \in S$ stop; a flow augmenting path has been found. (This is a path along which additional flow can be sent thereby increasing the total flow $F$). If not look for a pair $(i, j)$ with $i \in S$ and $j \in \bar{S}$ and either (i) $[(i, j) \in A$ and $f_{i,j} < u_{i,j}]$ or (ii) $[(j, i) \in A$ and $f_{j,i} > 0]$. If no such $(i, j)$ exists, then stop; $[f, F]$ is the optimal solution to the maximal flow problem and $(S, \bar{S})$ is a minimal cut separating $s$ and $t$. If $(i, j)$ of type (i) exists label $j(i^+, \epsilon_j)$ where $\epsilon_j = \min[\epsilon_i,\ u_{i,j} - f_{i,j}]$ and include $j$ in $S$. If $(i, j)$ is of type (ii) then label $j(i^-, \epsilon_j)$ where $\epsilon_j = \min[\epsilon_i,\ f_{j,i}]$ and include $j$ in $S$. Return to step 1. Type (i) arcs are called *forward* arcs and those of type (ii) are called *reverse* arcs. If we succeed in labeling $t$, we get an augmenting path as well as the nature of these arcs from the labels themselves. This is done as follows: If the label of $t$ is $(j^+, \epsilon_t)$ then the previous node to $t$ in the flow augmenting path is $j$ and the last arc is a forward arc; if it is $(j^-, \epsilon_t)$ then the previous node is still $j$ but the arc is a reverse arc. Now we look at the label of the node $j$ and the process is repeated until we reach $s$ and this identifies the entire path. We now augment the flow by $\epsilon_t$ along the path – by which we mean that flows on forward arcs along the path are increased by $\epsilon_t$ and those on reverse arcs are decreased by the same amount. This gives us a new feasible flow and the process is repeated. If the starting solution is optimal then at some point before labeling $t$ we will find no arc of type (i) or (ii). At this point the set $S$ of labeled nodes gives a minimum cut separating $s$ and $t$. Further, all arcs across the cut $(S, \bar{S})$ with the initial end in $S$ will be saturated (*i.e.* $f = u$) and all arcs with the terminal end in $S$ will be flowless $(f = 0)$. In other words, all forward arcs across the cut will be *saturated* and all reverse arcs will be *flowless*. Of course, if this happens with any feasible flow and any cut separating $s$ and $t$ then this flow is optimal and the corresponding cut is minimal.

Now we resume the main discussion of the primal dual algorithm for transportation and s assignment problems.

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 $\infty$. 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 $\delta$ 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.**