**LINEAR PROGRAMMING**

We consider Linear Programming problem in the following (standard) form. (where $A$ is an $m \times n$ matrix of rank $m$ , $c$ is $1 \times n$, $b$ is $m \times 1$ and $x$ is $n \times 1$) :

$$\text{Max } c \cdot x \qquad : \qquad Ax = b \qquad ; \qquad x \geq 0.$$

[Note : Inequality constraints can be converted to this form by adding "slack" variable. Also, we can do Gaussian Elimination on $A$ and if it does not have rank $m$, we either find that the system of equations has no solution, whence we may stop or we can find and discard redundent equations.]

A **basis** is a set of $m$ variables whose corresponding columns in $A$ are linearly independent; i.e., the $m \times m$ matrix $B$ formed by the columns corresponding to these variables is non-singular. These $m$ variables are called the "basic variables" (for this basis) and the rest $n - m$ variables are called "non-basic". The **basic solution** corresponding to this basis is the unique solution to the system of equations $Ax = b$ obtained by setting all the non-basic variables to 0. (So, the solution will be $B^{-1}b$.) If the basic solution is feasible (i.e., if all basic variables are non-negative), then it is called a basic feasible solution (bfs). The simplex algorithm moves from one bfs to another until we reach an optimal solution (there is always as optimal solution which is a bfs, as we will see).

Assume for the moment, that we have on hand a starting bfs. It will be convienient to rearrange the order of the variables, so the first $m$ variables are basic (for this bfs). Then the LP can be written as :

$$\begin{aligned} \text{Max } c_B x_B + c_N x_N & \\ B x_B + N x_N = b & \\ x_B, x_N \geq 0, & \end{aligned} \tag{1}$$

where we have partitioned $c$ into the basic part (subscripted by $B$) and the non-basic part and also with $x$. Noting that $B$ is invertible, the equations give us $x_B = B^{-1}b - B^{-1}N x_N$. Thus we may substitute this for $x_B$ and thus write LP as :

$$\begin{aligned} \text{Max } c_B B^{-1} b + (c_N - c_B B^{-1} N) x_N & \\ x_B = B^{-1}b - B^{-1}N x_N & \\ x_B, x_N \geq 0. & \end{aligned}$$

**Optimality Condition** : Now if

$$c_N - c_B B^{-1} N \leq 0 \tag{2}$$

then since in every feasible solution, the variables $x_N$ must all be non-negative and in the current bfs, they are all 0, we have an optimal solution. We may stop. Note that incidentally, we have found an optimal solution which is a bfs.

**A general step** : If a component of $c_N - c_B B^{-1} N$, say the coefficient of $x_j$ is positive, then the simplex algorithm will "bring the variable $x_j$ into the basis";

i.e., it will go to a new bfs with $x_j$ as a basic variable. Some other variable which is in the current bfs, must leave. To see which one, imagine increasing $x_j$ from its current value of 0. The question is how much we can increase $x_j$. We must maintain $x_B = B^{-1}b - B^{-1}Nx_N$; so as $x_j$ is increased, we must change also $x_B$; indeed $x_B$ is changed by the column of the matrix $B^{-1}N$ corresponding to $x_j$ multiplied by $x_j$. There are two possibilities :

**Unboundedness** : If the change in $x_B$ is by a non-negative vector, then we may increase $x_j$ without bound and so we may increase the objective function value without bound. In this case, we stop and say that the LP is unbounded. Note that the "change vector" gives us a direction $d$ with the following properties in this case :

$$Ad = 0 \quad ; \quad d \geq 0 \quad ; \quad c \cdot d > 0. \tag{3}$$

It is clear that if we have such a vector, the LP is unbounded. [The statement an "LP is unbounded" always means that if it is a Maximizatioin LP, then the objective value can be increased without upper bound remaining feasible. If it is a Minizatioin LP, then the objective value can be decreased without lower bound remaining feasible.]

**Moving to a better bfs** : In the other case, there is a largest value of $x_j$ such that beyond that one of the $x_B$ becomes negative. Then we increase $x_j$ to this value and put it into the basis. We change the $x_B$ accordingly and delete from the basis the variable among $x_B$ which has become 0. [It is a simple exercise to see that we have a new basis.]

If the increase in $x_j$ is strictly positive, then the objective function value has strictly inreased. So, we have made progress. If we always make such progress, then since the number of bfs 's is at most $\binom{n}{m}$, we will clearly terminate (albeit in exponentail number of steps in some cases). Unfortunately, this is not always the case and the algorithm can cycle.

**Avoiding Cycling** There are many methods for avoiding cycling. The simplest is the "perturbation method" in which the right hand side $b$ is perturbed by infinetismals; we add $\epsilon_1$ to $b_1$, $\epsilon_2$ to $b_2$, ........ $\epsilon_m$ to $b_m$ where

$$1 >> \epsilon_1 >> \epsilon_2 >> \ldots \epsilon_m.$$

[The $\epsilon$'s may be kept as symbols.] Now we claim that for each basis, the basic variables are all strictly positive. [This would suffice, because then the increase in $x_j$ is obviously strictly positive.] Thus it suffices to prove that $B^{-1}b'$ has no zero component for any basis matrix $B$, where $b'$ is the perturbed right hand side. If it did, then we will have that a row of $B^{-1}$ - say $v$ - times the vector $b'$ is zero. Since the perturbations are infitismal, (but the entries of $B^{-1}$ do not involve any infinitismals) this means that $v \cdot b = 0$. Thus $v \cdot (b' - b) = 0$. Since $\epsilon_2, \epsilon_3, \ldots$ are infitismally smaller than $\epsilon_1$, this implies that $v_1 = 0$. Now we must then have (since $\epsilon_3, \epsilon_4 \ldots$ are infinitsmally smaller than $\epsilon_2$) $v_2 =$ ..... Thus $v = 0$; but this contradcits the non-singularity of $B^{-1}$.

**Starting bfs** This is done by what is called Phase I. We first multiply the equations as necessary by -1, so $b$ has all non-ngeative components. We introduce a vector $y$ of $m$ artificial variables and first solve the LP :

$$\begin{aligned} \text{Max } & -\sum_{j=1}^{m} y_j \\ Ax + Iy &= b \\ x, y &\geq 0. \end{aligned}$$

For this problem, the $y$ 's form an obvious starting feasible basis.

**Lemma** The phase I LP is bounded. If the optimal solution value of the Phase I LP is 0, then we get a bfs for the original LP. if it is more than 0, then the original LP is infeasible.

**Theorem** If a LP has a finite optimal value, then the above method (upon termination) yields a bfs, which is an optimal solution to the LP.

**Duality**

Primal LP

$$\begin{aligned} \text{Max } & c \cdot x \\ Ax &= b \\ x &\geq 0. \end{aligned}$$

Dual LP

$$\begin{aligned} \text{Min } & y \cdot b \\ yA &\geq c \end{aligned}$$

**Lemma** (Weak Duality) : If $x$ is a feasible solution to the Primal LP and $y$ a feasible solution to the Dual LP, then we have

$$c \cdot x \leq y \cdot b.$$

**Proof** Since $x \geq 0$, and $yA \geq c$, we have $c \cdot x \leq yAx$. Since $Ax = b$, we have $yAx = yb$.

**Theorem** (Strong Duality) If the Primal LP has a finite optimal solution, then so does the dual LP and their values are equal. If the Primal LP is unbounded, then the Dual LP is infeasible. If the Primal LP is infeasible, then the Dual LP is unbounded or infeasible.

**Proof** If the Primal has a finite optimal solution, then it has a optimal bfs; wlg assume the LP is written in the form (1) with the basic vraibles of the optimal bfs as the first $m$ varibles. Consider the solution $y = c_B B^{-1}$. Clearly, $yA = (c_B, c_B B^{-1} N)$. From the optimality condition (2), we see that $y$ is feasible to the Dual LP. But we also have $yb = c_B B^{-1} b = c \cdot x$; so by weak Duality theorem, we have that $y, x$ are optimal to the Dual and Primal respectively proving the first part. If the primal is unbounded, then we have by (3), a vector $d$ with $Ad = 0$, $d \geq 0$ and $c \cdot d > 0$. If in addition, the Dual has a feasible solution $y$, then we would have $yAd = 0$, but $yAd \geq c \cdot d > 0$, a contradiction. So we get the second statement. To prove the third statement, we may take the dual of the Dual LP (after adding slack variables etc.) and we see that we get the Primal LP (Exercise). The applying the first statement, we get the third.