**The dual simplex method**

---

Convert the given linear programming problem in the following form:

$$(P) \qquad \min/\max z = C^t X + O^t s$$
$$\text{subject to} \qquad AX + Is = b, \qquad X, s \geq 0$$

Where $s = (s_1, s_2, ..., s_m)^t$ is a vector of slack variables and $O = (0,0...,0)_{n\times 1}$ is a vector of zeros. Also Assume that atleast one of the component $b_i$ of the RHS vector $b = (b_1, b_2, ..., b_m)$ is negative Initially define the following Input paramenters:

1. Enter the Matrix $A = [A\ I]$, where $I$ is an identity matrix of order m.
2. Entet the the R.H.S. vector $b$ and the cost matrix $C = [c\ O]_{(n+m)\times 1}$.
3. Define [m,n]=size (A)
4. Input the variables $s_1, s_2, ... s_m$ as initial basic variables.

Now construct the simplex table using $s_1, s_2, ..., s_m$ as initial basic variables. If the simplex table depicts **an optimal but Infeasible solution**, then dual simplex method is applicable. So apply the following procedure. 1. Select the leaving variable as $X_{B_r} = \min_i \{X_{B_i} \mid X_{B_i} < 0\}$

2. Select the entering variable $x_k$ using the formula $\dfrac{z_k - c_k}{y_{rk}} = \min_j \left\{ \dfrac{|z_j - c_j|}{|y_{rj}|} \ : \ y_{rj} < 0 \right\}$
3. Now update the basis as by removing $r^{th}$ basic variable with $k^{th}$ nonbasic variable. Again construct the simplex table and repeat the above procedure until an optimal basic feasible soluiton is not obtained.

**Least cost method of Transportation problem**

---

Here $a_i$ is the availability of the product at source $S_i$ and $b_j$ is the requirement of the same at destination $D_j$, $c_{ij}$ represents the cost of trasporting a unit product from source $S_i$ to destination $D_j$. The variable $x_{ij}$ is the quantity to be transported from the source $i$ to destination $j$.

Initially define Input paramenters:

1. Enter the number of sources as $m$, and destinations as $n$.
2. Enter the cost coefficients $c_{ij}$, the availabilty at $i^{th}$ source as $a_i$ and demand at $j^{th}$ destination as $b_j$ for each $i = 1,2...m$ , $j = 1,2,...n$.

Intially take $k = 1$

**Step 1:** Define $c_{pq} = \min(c_{ij})$, and assign $x_{pq} = \min(a_p,b_q)$, go to Step 2.

**Step 2:** If $\min(a_p,b_q) = a_p$, then update $b_q = b_q - a_p$, $a_p = a_p - x_{pq}$ else $\min(a_i,b_j) = b_q$, then update $a_p = a_p - b_q$, $b_q = b_q - x_{pq}$.

**Step 3** Assigne $c_{pq} = 10^5$ (a very large no.) , Set$k = k + 1$, if $k = m + n - 1$ go to Step 4 else go to Step 1.

**Step 4:** Stop and note the BFS and calculate the objective function value $z = \sum_{i,j} c_{ij}x_{ij}$

## Weighted Sum Method to multi-objective LPP

**Step 1:**Transform the multi-objective linear programming problem (P1) into its equivalent single objective linear programming problem (P2).

(P1)

Maximize/Minimize ($C_iX$), i=1,2,…,m

Subject to AX<=

or =or >= b X>=0.

(P1)

Maximize/Minimize ($C_1X + C_2X + \ldots + C_mX$)/m

Subject to AX<=

or =or >= b X>=0.

**Step 2:** Find an optimal solution of the problem i.e., an efficient solution of the problem (P1) by an appropriate method (Simplex method or Big-M method or Two-Phase method).

## Algorithm of Fibonacci Search Technique

**Step 1:** Using the relation, Measure of effectiveness $= \frac{Interval\ \ of\ uncertainty}{L_0}$, find the value of Measure of effectiveness.

**Step 2:** Using the relation $\frac{1}{F_n} \leq$ Obtained value of measure of effectiveness, find the smallest natural number $n$.

**Step 3:**Store the given interval [a,b]

**Step 3:** Find L0=b-a

**Step 4:**for i=n, find

$x_1 = a + \frac{F_{i-2}}{F_i} L_0,$ and $x_2 = a + \frac{F_{i-1}}{F_i} L_0,$

**Step 5:**If $f(x_1) > f(x_2)$ and the problem is of minimum. Then, repeat Step 3 with n=n-1a=$x_1$and . b=b

If $f(x_1) < f(x_2)$ and the problem is of minimum. Then, repeat Step 3 with n=n-1a=a and . b= x2.

If $f(x_1) > f(x_2)$ and the problem is of maximum. Then, repeat Step 3 with n=n-1a=a and . b=x2.

If $f(x_1) < f(x_2)$ and the problem is of maximum. Then, Then, repeat Step 3 with n=n-1a=$x_1$and . b=b

**Step 6:**Repeat Step 3 to upto i=2.