Thapar Institute of Engineering and Technology, Patiala
School of Mathematics

Optimization Techniques (UMA-035)
Lab Experiment- 6 (The dual simplex method)

The dual simplex method

Convert the given linear programming problem in the following form:

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

Where $s = (s_1, s_2, \ldots, s_m)^t$ is a vector of slack variables and $O = (0, 0 \ldots, 0)_{n\times 1}$ is a vector of zeros. Also Assume that atleast one of the component b_i of the RHS vector $b = (b_1, b_2, \ldots, b_m)$ 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 $s_1, s_2, \ldots s_m$ as initial basic variables.

Now construct the simplex table using $s_1, s_2, \ldots, s_m$ 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 $X_{B_r} = \min\limits_i \{X_{B_i} \mid X_{B_i} < 0\}$
2. Select the entering variable x_k using the formula $\dfrac{z_k - c_k}{y_{rk}} = \min_j \left\{ \dfrac{|z_j - c_j|}{|y_{rj}|} \quad : \ y_{rj} < 0 \right\}$
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.

Write a MATLAB code for the dual simplex method and test your program on the following examples:

1. $Min. \ \ z = 3x_1 + 5x_2, \ \ S.T. \ \ x_1 + 3x_2 \geq 3, \ x_1 + x_2 \geq 2, \ x_1, x_2 \geq 0.$
2. $Min. \ \ z = 12x_1 + 10x_2, \ \ S.T. \ \ 5x_1 + x_2 \geq 10, \ 6x_1 + 5x_2 \geq 30, \ x_1 + 4x_2 \geq 8, \ x_1, x_2 \geq 0.$
3. $min. \ \ z = 3x_1 + 2x_2, \ \ S.T. \ \ x_1 + x_2 \leq 1, \ x_1 + 2x_2 \geq 3, \ x_1, x_2 \geq 0.$

Thapar Institute of Engineering and Technology, Patiala
School of Mathematics

Optimization Techniques (UMA-035)
Lab Experiment- 7 (Least cost method)

Least cost method of Transportation problem

Consider a cost matrix representation of a Transportation problem:

				$u_i \downarrow$
c_{11} x_{11}	c_{12} x_{12}	$\cdots$	c_{1n} x_{1m}	a_1
c_{21} x_{21}	c_{22} x_{22}	$\cdots$	c_{2n} x_{2m}	a_2
$\vdots$				$\vdots$
c_{m1} x_{m1}	c_{m2} x_{m2}	$\cdots$	c_{mn} x_{mn}	a_m
b_1	b_2	$\cdots$	b_n	

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 $i = 1, 2 \ldots m$, $j = 1, 2, \ldots n$.

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 $z = \sum_{i,j} c_{ij} x_{ij}$

Write a MATLAB code to compute the basic feasible solutions of a Transportation problem using Northwest corner rule and test your program on the following set of examples:

1. Consider the cost matrix of the following transportation roblem

	D_1	D_2	D_3	D_4	a_i
S_1	2	10	4	5	12
$S_2 1$	6	12	8	11	25
S_3	3	9	5	7	20
b_j	25	10	15	5	

2. Consider the cost matrix of the following transportation roblem

	D_1	D_2	D_3	D_4	D_5	a_i
S_1	3	11	4	14	15	15
S_2	6	16	18	2	28	25
S_3	10	13	15	19	17	10
S_4	7	12	5	8	9	15
b_j	20	10	15	15	5	

Thapar Institute of Engineering and Technology, Patiala
School of Mathematics

Optimization Methods (UMA-035)
Lab Experiment - 8 (Multi-objective LPP)

Algorithm of 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 ($C_1X+ C_2X+…+ C_mX$)/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).

Write a MATLAB code to solve the following multi-objective LPPs by weighted sum method:

1. Maximize ($3x_1 + 2x_2+4x_3$)
   Maximize ($x_1 + 5x_2+3x_3$)
   Subject to
   $2x_1 +4x_2 +x_3 <= 8$,
   $3x_1 + 5x_2 +4x_3 >= 15$,
   $x_1>=0, x_2>= 0, x_3>= 0$

2. Maximize ($x_1 + 4x_2+x_3$)
   Maximize ($2x_1 + 7x_2+5x_3$)
   Subject to
   $x_1+x_2 +x_3 <= 8$,
   $x_1+ 5x_2 +4x_3 >= 15$,
   $x_1>=0, x_2>= 0, x_3>= 0$

Thapar Institute of Engineering and Technology, Patiala
School of Mathematics

Optimization Methods (UMA-035)
Lab Experiment - 9 (Fibonacci Search Technique)

Algorithm of Fibonacci Search Technique

Fibonacci numbers

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
1	1	2	3	5	8	13	21	34

$F_0 = 1, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n > 1.$

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 $L_0 = b - a$

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 $n = n - 1 a = x_1$and $b = b$.

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

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

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

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

Write a MATLAB code to solve the following problem

Example:

Minimize the function $x(x - 2)$, $0 \leq x \leq 1.5$ within the interval of uncertainty $0.25L_0$.