Optimisation Techniques (UMA035)

Submitted To:

Submitted By

Granth Dhir
102003363

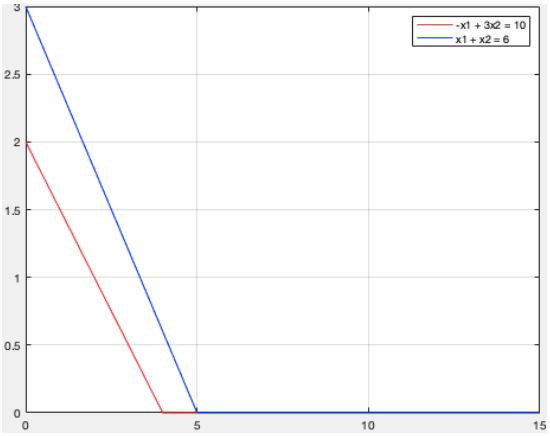
February 2023

Experiment 1 (Graphical Method)

```
%graphical method
%how to plot lines in matlab
% -x1 + 3x2 = 10
% x1 + x2 = 6
% x1 - x2 = 2
clc
clear all
format short
%phase-1 : input parameters
A = [2 4; 3 5];
B = [8; 15];

%phase-2 : to plot the lines on the graph
y1 = 0:max(B);
x11 = (B(1) - A(1,1).*y1 )./A(1,2);
x21 = (B(2) - A(2,1).*y1 )./A(2,2);
%  x31 = (B(3) - A(3,1).*y1 )./A(3,2);
%to ignore negative portion
x11 = max(0,x11);
x21 = max(0,x21);
% x31 = max(0,x31);

plot(y1,x11,'r',y1,x21,'b')
title('graph of x1 vs x2')
xlabel('values of y1')
ylabel('values of x2')
legend('-x1 + 3x2 = 10','x1 + x2 = 6','x1 - x2 = 2')
grid on
```

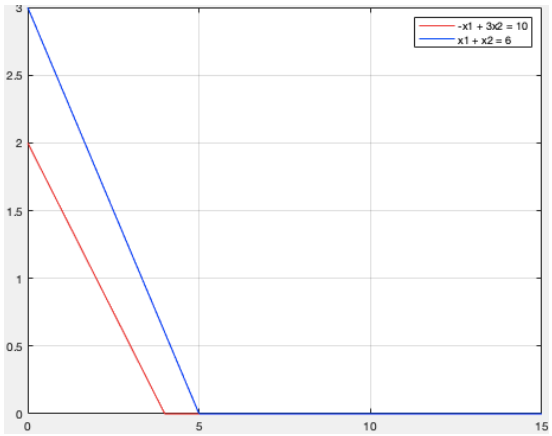
```
%phase-3 : to find the corner points with axes
cx1 = find(y1==0);
c1 = find(x11==0);
line1 = [y1(:,[c1,cx1]);x11(:,[c1,cx1])]'     % ' for transpose
c2 = find(x21==0);
line2 = [y1(:,[c2,cx1]);x21(:,[c2,cx1])]'
% c3 = find(x31==0);
% line3 = [y1(:,[c3,cx1]);x31(:,[c3,cx1])]'

corpt = unique([line1;line2],'rows')         %set of corner pts

%phase-4 : to find the points of intersection of all the lines
pt = [0;0]
for i=1:size(A,1)
    A1 = A(i,:);
    B1 = B(i,:);
    for j=i+1:size(A,1)
        A2 = A(j,:);
        B2 = B(j,:);
        A4 = [A1;A2];
        B4 = [B1;B2];

        x = A4\B4;
        pt = [pt,x];
    end
end
ptt = pt'
% points = [0,0; 0,3.3333; 0,6; 1,0; 2,0; 2,4; 4,2; 6,0; 7,0; 8,0;
8,6; 9,0; 10,0]
```

```
% phase-5 : write all corner pts
allpt = [ptt;corpt];
points = unique(allpt,'rows');

% phase-6 : to check the feasibility
PT = constraint(points);
P = unique(PT,'rows')
%phase-7 :  to find objective func
C = [1 5];
for i=1:size(P,1)
    fn(i,:) = sum(P(i,:).*C);
end
ver_fns = [P fn];

% phase-8 : to find the optimal soln
[optval optposition] = max(fn);
optval = ver_fns(optposition,:);
OPTIMAL_BFS = array2table(optval)
OPTIMAL_BFS.Properties.VariableNames(1:size(optval,2)) =
{'x1','x2','z'}
```



x1	x2	z
0	3	15

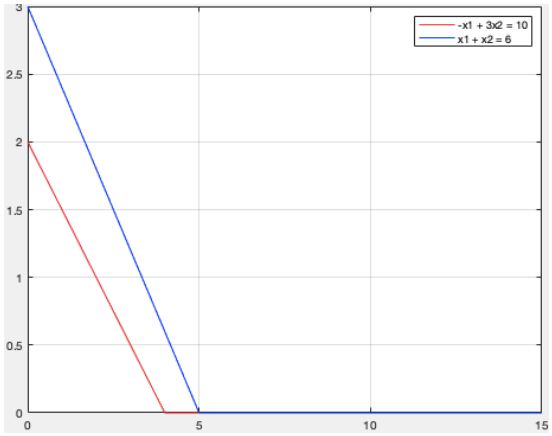
Q2



x1	x2	z
15	0	45

x1	x2	z
0	3	6

Q3



x1	x2	z
10	-3	24

Lab Experiment- 2 (Basic feasible solutions)

Code:

```
clc
clear all
% max z = 2x1+3x2+4x3+7x4
% st   2x1+3x2-x3+4x4 = 8
%      x1-2x2+6x3-7x4 = -3
%      x1,x2,x3,x4>=0
C = [2 3 4 7]
A = [2 3 -1 4; 1 -2 6 -7]
b = [8; -3]
s = eye(size(a,1),size(I,2))
idx = find(I>0)
s(idx,idx) = -s(idx,idx)
mat = [a s b]
ans = array2table(mat)
ans.Properties.VariableNames(1:size(mat,2)) =
{'x1','x2','x3','s1','s2','s3','max'}
```

x1	x2	x3	s1	s2	s3	max
1	-2	-3	1	0	0	4
2	3	-4	0	-1	0	5
1	1	1	0	0	-1	2

```
n = size(A,2)
m = size(A,1)
if (n>m)
    ncm = nchoosek(n,m)
    pair = nchoosek(1:n,m)
```

```
    sol = []
    for i=1:ncm
        y = zeros(n,1)
        x = A(:,pair(i,:))\b
        if all(x>=0 & x~=inf & x~=-inf)     %extracting BFS
            y(pair(i,:)) = x
            sol = [sol y]
        end
    end

else
    disp('No feasible region')
end

% to find objective function value
z = C*sol
[zmax, zindex] = max(z)
bfs2 = sol(:,zindex)
optbfs = [bfs2' zmax]
optimalbfs = array2table(optbfs)
optimalbfs.Properties.VariableNames(1:size(optbfs,2)) =
{'x1','x2','x3','x4','max'}
```

x1	x2	x3	x4	max
0	0	2.5882	2.6471	28.882

Q1

Command Window

x1	x2	s1	s2	max
0.5	1.5	0	0	3.5

>>

Q2

Command Window

No Feasible Region

>>

Q3

Command Window

x1	x2	x3	x4	s1	s2	s3	max
0	4	0	4	0	2	4	8

Q4

Command Window

x1	x2	x3	s1	s2	max
0	1	1	0	0	2