Name : Garvit Nangia Roll No. : 101903778 Group : 3COE2

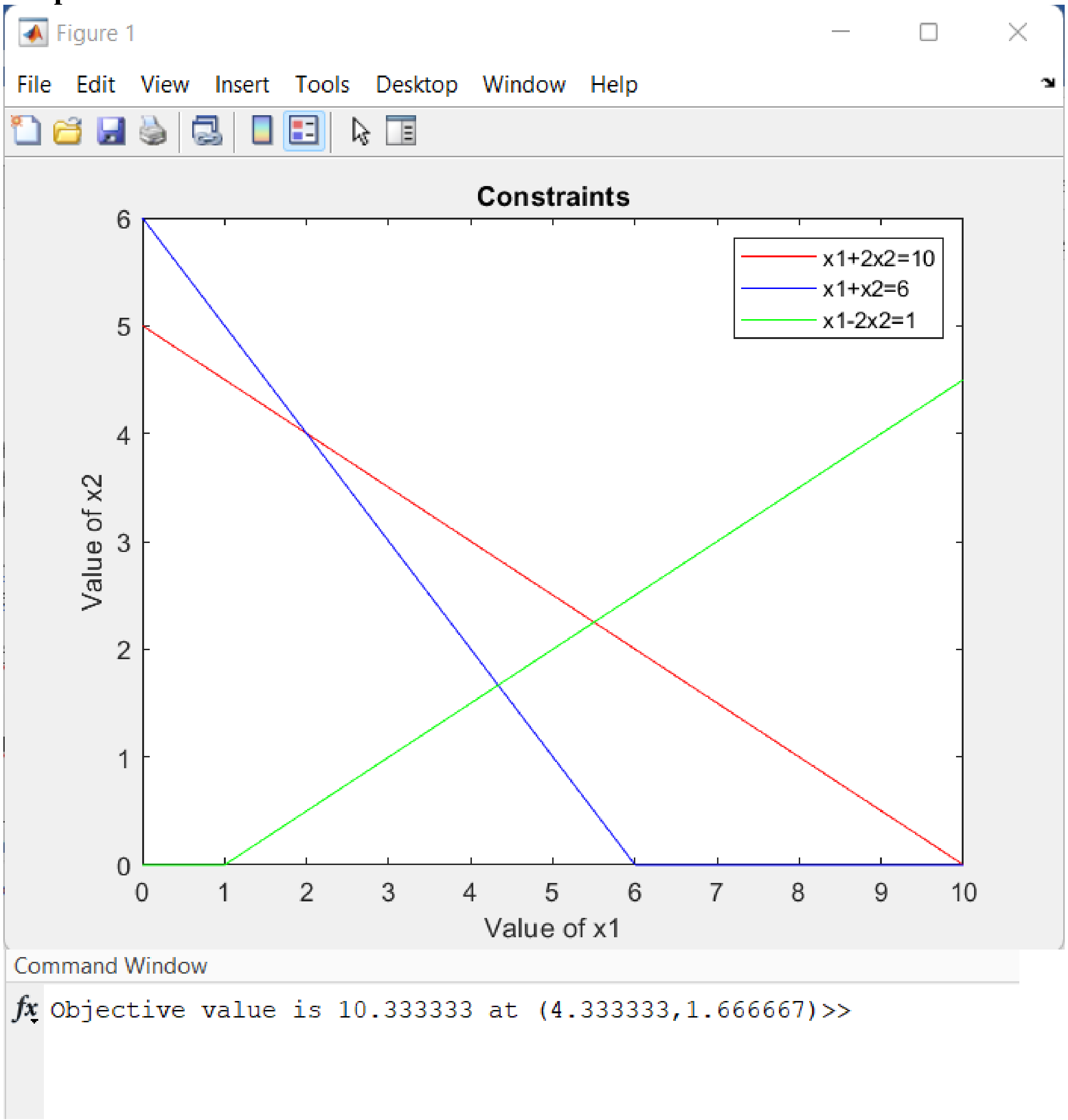
Optimization Techniques

Assignment 1: Graphical Method

```
clc;
clear all;

a=[1 2;1 1;1 -2];
b=[10;6;1];
c=[2 1];

p=max(b);
x1=0:p;
x21 = (b(1)-a(1,1)*x1)/a(1,2);
x22 = (b(2)-a(2,1)*x1)/a(2,2);
x23 = (b(3)-a(3,1)*x1)/a(3,2);

x21=max(x21,0);
x22=max(x22,0);
x23=max(x23,0);

plot(x1,x21,'r',x1,x22,'b',x1,x23,'g');
title('Constraints')
xlabel('Value of x1')
ylabel('Value of x2')
legend('x1+2x2=10','x1+x2=6','x1-2x2=1')

%find returns position
cx1=find(x1==0);
c1=find(x21==0);
c2=find(x22==0);
c3=find(x23==0);

line1=[x1([cx1 c1]);x21([cx1 c1])]';
line2=[x1([cx1 c2]);x22([cx1 c2])]';
line3=[x1([cx1 c3]);x23([cx1 c3])]';

corner = unique([line1;line2;line3],'rows');
pt = [0;0];
for i = 1:size(a,1)
    for j = i+1:size(a,1)
        a1 = a([i j],:);
        b1 = b([i j],:);
        x = inv(a1)*b1;
        pt = [pt x];
    end
end

pt = pt';

points = unique([corner;pt],'rows');
for i=1:size(points,1)
    const1(i) = a(1,1)*points(i,1)+a(1,2)*points(i,2)-b(1);
    const2(i) = a(2,1)*points(i,1)+a(2,2)*points(i,2)-b(2);
    const3(i) = a(3,1)*points(i,1)+a(3,2)*points(i,2)-b(3);
```

```
end

s1 = find(const1>0);
s2 = find(const2>0);
s3 = find(const3>0);

s = unique([s1 s2 s3]);
points(s,:) = [];
value = points*c';
[obj,index] = max(value);
sol = points(index,:);

fprintf('Objective value is %f at (%f,%f)',obj,sol(1),sol(2));
```

Output:

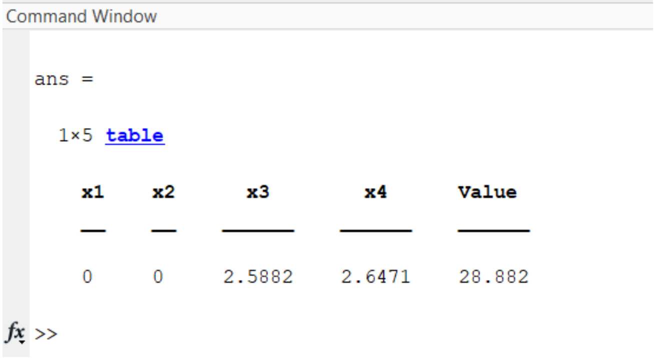


```
Objective value is 10.333333 at (4.333333,1.666667)>>
```

Assignment 2: Basic Feasible Solution

```
clc;
clear all;

a = [2 3 -1 4; 1 -2 6 -7];
b = [8;-3];
c = [2 3 4 7];

m = size(a,1);
n = size(a,2);

maxSoln = nchoosek(n,m); % no of combinations : nCm
p = nchoosek(1:n,m); % series of solns

if(n>=m)
    basicSoln = [];
    for i = 1:maxSoln
        a1 = a(:,p(i,:));
        x = inv(a1)*b;
        if x>=0 & x~=inf & x~=-inf
            y = zeros(n,1);
            y(p(i,:)) = x;
            basicSoln = [basicSoln y];
        end
    end
    val = c*basicSoln;
    [obj,index] = max(c*basicSoln);
    bfs = basicSoln(:,index)';
    optimal = [bfs obj];
    array2table(optimal,'VariableNames',{'x1','x2','x3','x4','Value'})
else
    error("Error, no of variables must be greater than no of equations");
end
```

Output:



```
ans =

  1×5 table

    x1    x2      x3        x4       Value 
    __    __    ______    ______    ______

    0     0     2.5882    2.6471    28.882

>>
```