MATH 415: Assignment 6

Md Ismail Hossain

31st October 2022

Question 1:

The problem given as:

$$\begin{aligned}
\text{minimize} \quad & z = -5x_1 - x_3 \\
\text{subject to} \quad & x_1 + x_2 + x_3 \leq 1, \\
& 2x_1 + x_2 \leq 2, \\
& x_1 + 2x_2 + 3x_3 \leq 3, \\
& \mathbf{x} \geq 0
\end{aligned}$$

We need to write down the dual of the LP. To do that we have to convert this primal LP into a standard form:

$$\begin{aligned}
\text{maximize} \quad & z = 5x_1 + x_3 \\
\text{subject to} \quad & x_1 + x_2 + x_3 \leq 1, \\
& 2x_1 + x_2 \leq 2, \\
& x_1 + 2x_2 + 3x_3 \leq 3, \\
& \mathbf{x} \geq 0
\end{aligned}$$

Now the dual of this LP can be written as:

$$\begin{aligned}
\text{minimize} \quad & z = y_1 + 2y_2 + 3y_3 \\
\text{subject to} \quad & y_1 + 2y_2 + y_3 \geq 5, \\
& y_1 + y_2 + 2y_3 \geq 0, \\
& y_1 + 3y_3 \geq 1, \\
& \mathbf{y} \geq 0
\end{aligned}$$

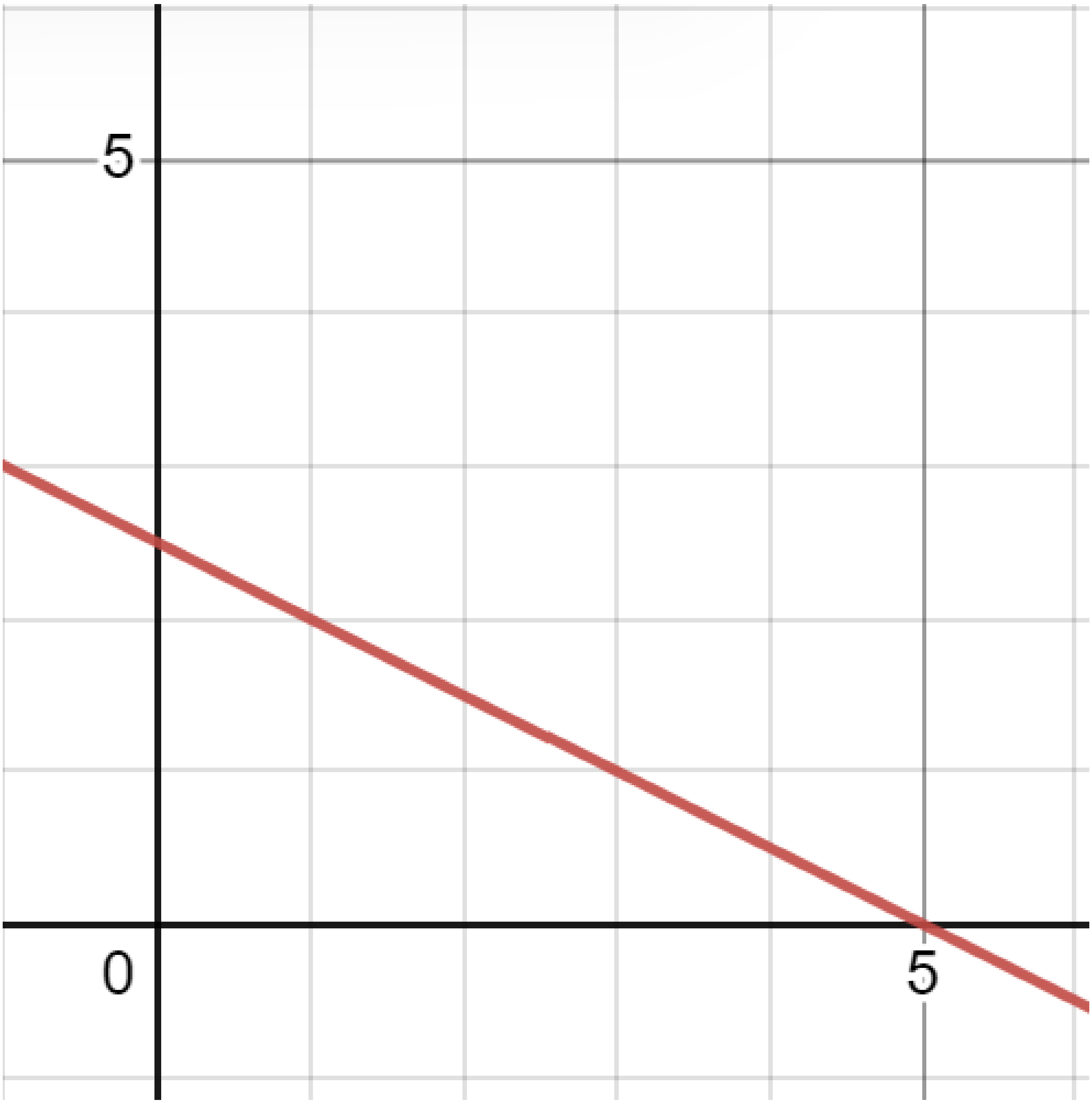
Now, we need to show $x_1 = 1, x_2 = 0, x_3 = 0$ is an optimal solution. To do that we need to check whether the constraints of the primal LP are valid for this solution or not.

The first constraints of the primal LP can be verified as: $1 + 0 + 0 = 1$ which is equal to 1. As a result, the corresponding slack will be 0. Similarly, the other two constraints are, $2(1) + 0 = 2$ exactly equal to the right side (corresponding slack 0) and $1 + 0 + 0 = 1 < 3$ (corresponding slack 1). So, we can say the given solution satisfies the constraints and the slack variables are $(0, 0, 1)$ for the original LP.

According to complementary slackness y_1, y_2 & y_3 are the optimal solutions for the dual, and as the slack for the original LP is $(0, 0, 1)$ then y_3 must be 0 by theorem. Now we need to find the solution for y_1 and y_2.

The solution set for the primal LP is $(x_1, x_2, x_3) = (1, 0, 0)$, as only the first solution is greater than 0, we have to consider the first constraints of the dual as: $y_1 + 2y_2 = 5$ and the other two constraints of the dual would not be considered because corresponding solutions of the original variables are not greater than 0.

Now have another challenge there are two unknown variables and only one equation. As a result, there are infinitely many solutions that could be available. If observe the graph of this equation then we can make a decision about the range of valid values for y_1 and y_2.



So, the solution for the dual is:
$0 \leq y_1 \leq 5$, $0 \leq y_2 \leq 2.5$ and $y_3 = 0$

Question 2:

```
    % Given Information
A = [-1 -1 2 1 0;1 -1 -1 0 1]

A =

    -1    -1     2     1     0
     1    -1    -1     0     1

b = [-1;-5]

b =

    -1
    -5

c = [2;1;3;0;0]

c =

     2
     1
     3
     0
     0

u = [Inf; Inf; Inf; Inf;Inf]

u =

   Inf
   Inf
   Inf
   Inf
   Inf



%Optimal basis for this problem is x2, x4

basis = [2 4]

basis =

     2     4

nonbasis0 = [1 3 5]

nonbasis0 =

     1     3     5

nonbasisu = []

nonbasisu =

     []
```

```
basisupdateu

x =

     0
     5
     0
     4
     0


z =

     5


reducedatzero =

     3     2     1
     1     3     5


reducedatupper =

     []



%Required values
xstar = xb

xstar =

     5
     4

ystar = y

ystar =

     0
    -1

z

z =

     5




%Check the Ranging and sensitivity analysis of b(1)


ei = [1;0]
```

```
ei =

     1
     0


d = solveBxb(L,U,p,-ei)

d =

     0
    -1




eps2 = 1.0e-5; eps3 = 1.0e-5;

%Upper limit for delb
[tlimit,leavingvar,leavingbound] = ratiotestu(basis,xb,ub,d,+Inf,eps2,eps3)

tlimit =

   Inf


leavingvar =

     0


leavingbound =

     1

upperlimit = tlimit

upperlimit =

   Inf


%Lower limit for delb
[tlimit,leavingvar,leavingbound] = ratiotestu(basis,xb,ub,-d,+Inf,eps2,eps3)

tlimit =

     4


leavingvar =

     4


leavingbound =

     0
```

```
lowerlimit = tlimit

lowerlimit =

     4


%Limit of b1
[b(1)- lowerlimit b(1)+ upperlimit]

ans =

    -5   Inf

% Range for b(1) is [-5, Inf]

%Will be optimal because rn0 does not depend on b

% Sensitivity

y1star = y(1)

y1star =

     0

%As y1star is 0; sensitivity will be 0 if we change b1 between -5 to Inf.

%%%%%%%%%%%%%%%%%%%%%%%%%%%
c(1)-rn0(1)

ans =

    -1

%So, range for c1 is [-1 Inf] and sensitivity is 0 within the ranges, because c1 is nonbasis0.
```

Question 3:

```
A = [-1 -1 0 1 0;1 -1 -1 0 1]

A =

    -1    -1     0     1     0
     1    -1    -1     0     1

b = [-2;-3]

b =

    -2
    -3

c = [5;3;2;0;0]

c =

     5
     3
     2
     0
     0

u = Inf*ones(1,5)

u =

   Inf   Inf   Inf   Inf   Inf


basis = [4 5]

basis =

     4     5

nonbasis0 = [1 2 3]

nonbasis0 =

     1     2     3

nonbasisu = []

nonbasisu =

     []


basisupdateu;

x =

     0
```

```
     0
     0
    -2
    -3


z =

     0


reducedatzero =

     5     3     2
     1     2     3


reducedatupper =

     []



B=A(:,basis)

B =

     1     0
     0     1

A(:,nonbasis0)

ans =

    -1    -1     0
     1    -1    -1

A(:,nonbasisu)

ans =

  2×0 empty <a href="matlab:helpPopup double" style="font-weight:bold">double</a> matrix

cb=c(basis)

cb =

     0
     0

cn0=c(nonbasis0)

cn0 =

     5
     3
     2
```

```
cnu=c(nonbasisu)

cnu =

     []

xb

xb =

    -2
    -3

y

y =

     0
     0

rn0

rn0 =

     5     3     2

rnu

rnu =

     []




%x5 leaves the basis becasue it's the most negative one. As all reduced
%atzero are greater than 0 then we use dual simplex method
% k = 2

ek = [0;1]

ek =

     0
     1

v = ek'*inv(B)*A(:,nonbasis0)

v =

     1    -1    -1

ratios = v./rn0

ratios =

    0.2000   -0.3333   -0.5000
```

```
%x3 going to enter the basis becasue it’s ratio is most negative

basis=[3 4]

basis =

     3     4

nonbasis0 = [1 2 5]

nonbasis0 =

     1     2     5

basisupdateu;

x =

     0
     0
     3
    -2
     0


z =

     6


reducedatzero =

     7     1     2
     1     2     5


reducedatupper =

     []


B=A(:,basis)

B =

     0     1
    -1     0

A(:,nonbasis0)

ans =

    -1    -1     0
     1    -1     1

A(:,nonbasisu)
```

```
ans =

  2×0 empty <a href="matlab:helpPopup double" style="font-weight:bold">double</a> matrix

cb=c(basis)

cb =

     2
     0

cn0=c(nonbasis0)

cn0 =

     5
     3
     0

cnu=c(nonbasisu)

cnu =

     []

xb

xb =

     3
    -2

y

y =

     0
    -2

rn0

rn0 =

     7     1     2

rnu

rnu =

     []


%x4 going to leave, k  = 2

v = ek'*inv(B)*A(:,nonbasis0)

v =
```

```
    -1    -1     0

ratios = v./rn0

ratios =

   -0.1429   -1.0000         0


%x2 going to enter
basis=[2 3]

basis =

     2     3

nonbasis0 = [1 4 5]

nonbasis0 =

     1     4     5

basisupdateu;

x =

     0
     2
     1
     0
     0


z =

     8


reducedatzero =

     6     1     2
     1     4     5


reducedatupper =

     []


B=A(:,basis)

B =

    -1     0
    -1    -1

A(:,nonbasis0)
```

```
ans =

    -1     1     0
     1     0     1

A(:,nonbasisu)

ans =

  2×0 empty <a href="matlab:helpPopup double" style="font-weight:bold">double</a> matrix

cb=c(basis)

cb =

     3
     2

cn0=c(nonbasis0)

cn0 =

     5
     0
     0

cnu=c(nonbasisu)

cnu =

     []

xb

xb =

     2
     1

y

y =

    -1
    -2

rn0

rn0 =

     6     1     2

rnu

rnu =

     []
```

```
% Checking

A'*y-c

ans =

    -6
     0
     0
    -1
    -2


% As all entry less than or equal to 0 ... we are fine. Optimal solution z
% = 8.
```