MECH6318 HW04

Monday, September 27, 2021 9:09 PM

Fall 2021 | MECH 6318
Engineering Optimization – Prof. Jie Zhang
HOMEWORK #4
September 21, 2021

DUE: Tuesday, September 28, 2021 **5pm (central time)**
Submit the HW to eLearning

Points Distribution
30 points maximum
-5 to 0 points reserved for **Neatness and Professional Presentation**
(legible, stapled, show key Matlab commands, properly labeled plots, etc.)

Matlab Problem:

Write a Matlab code for the Simplex method for linear programming that estimates the minimum of the function in Problem 11.8. Verify the correctness of your solution from HW3.

Additional Problem:

Consider the following LP Problem:

$$\text{Minimize } f = 3x_1 + x_3 + 2x_5$$

Subject to

$$x_1 + x_3 - x_4 + x_5 = -1$$
$$x_2 - 2x_3 + 3x_4 + 2x_5 = -2$$
$$x_i \geq 0, \; i = 1 \text{ to } 5$$

Solve this problem using the dual simplex method.

Problem 1

11.8 Solve the following problem using the Simplex method. Verify the correctness of your solution using `linprog`.

$$\min_{x} \; x_1 + 2x_2 - 7x_3 \tag{11.135}$$

subject to

$$2x_1 + x_2 + x_3 \leq 15 \tag{11.136}$$
$$-x_1 + 2x_2 - x_3 \leq 7 \tag{11.137}$$
$$x_1 + 5x_2 + 5x_3 \leq 25 \tag{11.138}$$
$$x_1, x_2, x_3 \geq 0 \tag{11.139}$$

Simplex code:

```matlab
% Simplex Algorithm Function
function [x_min, f_min, n_iter, T] = simplex(f,A,b)
    %   Syntax  : [x_min, f_min, n_iter, T] = simplex(f,A,b)
    %
    %
    %   Purpose : Solves the problem
    %
    %               min      f'*x
    %               st.      A*x    <= b
    %                        x >= 0
    %   Assumes no equality constraints and x_i >= 0 forall i
    arguments
        f (:,1) double {mustBeNumeric,mustBeReal}
        A (:,:) double {mustBeNumeric,mustBeReal} = []
        b (:,1) double {mustBeNumeric,mustBeReal} = []
    end
    % Assuming everything inputed is good....
    max_iter = 20;
    % Setup
    n = size(f,1);
    num_s = size(A,1);% 0;

    T = [[A;f'],eye(size(A,1)+1),[b;0]];
    disp('Initial T = ');
    disp(T);
```

```matlab
n_iter = 0;
ratios = zeros(size(T,1)-1,1);
while any(T(end,1:end-1)<0) %Keeps going until optimal (final row >= 0)
    n_iter = n_iter + 1;
    % complicated way to find smallest value index
    min_val = min(T(end,1:end-1));
    [~, min_col] = find(T(end,1:end-1)==min_val,1,'first');
    % find pivot row
    for row = 1:(size(T,1)-1)
        if T(end,row) >= 0
            if T(row, min_col) > 0
                ratios(row) = T(row,end) / T(row,min_col);
            else
                ratios(row) = inf;
            end
        else
            ratios(row) = inf;
        end
    end
    min_val = min(ratios);
    [min_row,~] = find(ratios==min_val,1,'first');

    % Pivoting
    new_T = zeros(size(T));
    new_T(min_row,:) = T(min_row,:)/T(min_row,min_col);
    for row = 1:size(T,1)
        if row ~= min_row
            new_T(row,:) = T(row,:) ...
                - T(row,min_col) * new_T(min_row,:);
        end
    end
    T = new_T;

    if n_iter >= max_iter
        error('too many iterations')
    end

end
```

```matlab
    disp('Final T = ');
    disp(T);


    % Calculating the basic variables
    j = 1;
    row = zeros(size(T,1),1);
    col = zeros(size(T,1),1);
    for i = 1:size(T,2)
        if nnz(T(:,i)) == 1
            col(j) = i;
            row(j) = find(T(:,i),1);
            j = j+1;
        end
    end

    % Solving for x values
    X = zeros([n,n+1]);
    for i = 1:n
        if col(i) <= n
            X(i,col(i)) = 1;
            X(i,end) = T(i,end);
        else
            X(i,:) = [A(i,:),b(i) - T(i,end)] ;
        end
    end
    X = rref(X);
    x_min = X(:,end);
    f_min = f'*x_min;
end
```

Result:

```
% Problem 1  ---------------------------------------

% Wrote a seperate script: simplex.m

% Problem 11.8 Problem
f = [1;
    2;
    -7]

A = [2, 1, 1;
    -1, 2,-1;
    1, 5, 5]
b = [15;
    7;
    25]


[x_min, f_min, n_iter,T] = simplex(f,A,b)
```

```
f = 3×1
     1
     2
    -7

A = 3×3
     2     1     1
    -1     2    -1
     1     5     5

b = 3×1
    15
     7
    25

Initial T =
     2     1     1     1     0     0     0    15
    -1     2    -1     0     1     0     0     7
     1     5     5     0     0     1     0    25
     1     2    -7     0     0     0     1     0

Final T =
     2     1     1     1     0     0     0    15
     1     3     0     1     1     0     0    22
    -9     0     0    -5     0     1     0   -50
    15     9     0     7     0     0     1   105
x_min = 3×1
     0
     0
    15

f_min = -105
n_iter = 1
T = 4×8
     2     1     1     1     0     0     0    15
     1     3     0     1     1     0     0    22
    -9     0     0    -5     0     1     0   -50
    15     9     0     7     0     0     1   105
```

Comparison... I think last week's assignment was really really wrong...

```
final_simplex_tbl = T
```

final_simplex_tbl =

$$\begin{pmatrix} 1 & 0 & 0 & \frac{5}{9} & 0 & -\frac{1}{9} & \frac{50}{9} \\ 0 & 1 & 0 & \frac{4}{27} & \frac{1}{3} & \frac{1}{27} & \frac{148}{27} \\ 0 & 0 & 1 & -\frac{7}{27} & -\frac{1}{3} & \frac{5}{27} & -\frac{43}{27} \\ 0 & 0 & 0 & -\frac{8}{3} & -3 & \frac{4}{3} & f-\frac{83}{3} \end{pmatrix}$$

```
final_simplex_soln = T(1:3,7)
```

final_simplex_soln =

$$\begin{pmatrix} \frac{50}{9} \\ \frac{148}{27} \\ -\frac{43}{27} \end{pmatrix}$$

```
final_simplex_value = c'*final_simplex_soln
```

final_simplex_value =

$$\frac{83}{3}$$

Problem 2

Additional Problem:

Consider the following LP Problem:

$$\text{Minimize } f = 3x_1 + x_3 + 2x_5$$

Subject to

$$x_1 + x_3 - x_4 + x_5 = -1$$
$$x_2 - 2x_3 + 3x_4 + 2x_5 = -2$$
$$x_i \geq 0,\ i = 1 \text{ to } 5$$

Solve this problem using the dual simplex method.

Problem 2

```
disp('Problem 2 ----------------------------')
% Problem setup
f = [3;
     0;
     1;
     0;
     2]

A = [];
b = [];
C = [1, 0, 1,-1, 1;
     0, 1,-2, 3, 2]
d = [-1;
     -2]
l = 0;
u = inf;

n = size(f,1);
m = size(C,1);

% Dual Simplex Method

% Positive right side...
A = diag(sign(d)) * C
b = diag(sign(d)) * d

% Fake State variables
Y = eye(m)

% Table Construct
T = [[A,            eye(m),     zeros(m,1), b];
      f',           zeros(1,m), 1,          0;
      zeros(1,n),   ones(1,m),  1,          0]
for i = 1:m
    T(end,:) = T(end,:) - T(i,:);
end
T
```

```
Problem 2 ----------------------------
f = 5×1
     3
     0
     1
     0
     2

C = 2×5
     1     0     1    -1     1
     0     1    -2     3     2

d = 2×1
    -1
    -2

A = 2×5
    -1     0    -1     1    -1
     0    -1     2    -3    -2

b = 2×1
     1
     2

Y = 2×2
     1     0
     0     1

T = 4×9
    -1     0    -1     1    -1     1     0     0     1
     0    -1     2    -3    -2     0     1     0     2
     3     0     1     0     2     0     0     1     0
     0     0     0     0     0     1     1     1     0

T = 4×9
    -1     0    -1     1    -1     1     0     0     1
     0    -1     2    -3    -2     0     1     0     2
     3     0     1     0     2     0     0     1     0
     1     1    -1     2     3     0     0     1    -3
```

```
% Phase 1
min_col = 3; %1,1,-1,2,3,0,0,1
min_row = 3; %Ratios:inf,1,0

% Pivoting
new_T = zeros(size(T));
new_T(min_row,:) = T(min_row,:)/T(min_row,min_col);
for row = 1:size(T,1)
```

```
T = 4×9
     2     0     0     1     1     1     0     1     1
    -6    -1     0    -3    -6     0     1    -2     2
     3     0     1     0     2     0     0     1     0
```

```
% Pivoting
new_T = zeros(size(T));
new_T(min_row,:) = T(min_row,:)/T(min_row,min_col);
for row = 1:size(T,1)
    if row ~= min_row
        new_T(row,:) = T(row,:) ...
            - T(row,min_col) * new_T(min_row,:);
    end
end
T = new_T

% Optimal solution...
x3 = 0;
y1 = 1;
y2 = 2;

w = y1 + y2
if w > 0
    disp('w > 0, infeasable')
end

% Confirmation:
disp('Confirmaiton with Linprog:')
[~,~,exitflag]=linprog(f,[],[],C,d,zeros(n,1))
disp('exitflag = -2 => no feasable solution found')
```

```
T = 4×9
     2     0     0     1     1     1     0     1     1
    -6    -1     0    -3    -6     0     1    -2     2
     3     0     1     0     2     0     0     1     0
     4     1     0     2     5     0     0     2    -3

w = 3

w > 0, infeasable

Confirmaiton with Linprog
exitflag = -2
exitflag = -2 => no feasable solution found
```

```
% MECH 6318 - HW 2
% Jonas Wagner
% 2021-09-07

clear
close all
```

Problem 1

```
disp('Problem 1 ----------------------------')
```

```
Problem 1 ----------------------------
```

```
% Wrote a seperate script for algorithm: simplex.m

% Problem 11.8 setup
f = [1;
     2;
     -7]
```

```
f = 3×1
     1
     2
    -7
```

```
A = [2, 1, 1;
     -1, 2,-1;
     1, 5, 5]
```

```
A = 3×3
     2     1     1
    -1     2    -1
     1     5     5
```

```
b = [15;
     7;
     25]
```

```
b = 3×1
    15
     7
    25
```

```
% Simplex Algorithm
[x_min, f_min, n_iter,T] = simplex(f,A,b)
```

```
Initial T =
     2     1     1     1     0     0     0    15
    -1     2    -1     0     1     0     0     7
     1     5     5     0     0     1     0    25
     1     2    -7     0     0     0     1     0

Final T =
     2     1     1     1     0     0     0    15
```

```
     1     3     0     1     1     0     0    22
    -9     0     0    -5     0     1     0   -50
    15     9     0     7     0     0     1   105
x_min = 3×1
     0
     0
    15
f_min = -105
n_iter = 1
T = 4×8
     2     1     1     1     0     0     0    15
     1     3     0     1     1     0     0    22
    -9     0     0    -5     0     1     0   -50
    15     9     0     7     0     0     1   105
```

Problem 2

```
disp('Problem 2 -----------------------------')
```

```
Problem 2 -----------------------------
```

```
% Problem setup
f = [3;
     0;
     1;
     0;
     2]
```

```
f = 5×1
     3
     0
     1
     0
     2
```

```
A = [];
b = [];
C = [1, 0, 1,-1, 1;
     0, 1,-2, 3, 2]
```

```
C = 2×5
     1     0     1    -1     1
     0     1    -2     3     2
```

```
d = [-1;
     -2]
```

```
d = 2×1
    -1
    -2
```

```
l = 0;
u = inf;

n = size(f,1);
m = size(C,1);
```

```
% Dual Simplex Method

% Positive right side...
A = diag(sign(d)) * C
```

```
A = 2×5
    -1     0    -1     1    -1
     0    -1     2    -3    -2
```

```
b = diag(sign(d)) * d
```

```
b = 2×1
     1
     2
```

```
% Fake State variables
Y = eye(m)
```

```
Y = 2×2
     1     0
     0     1
```

```
% Table Construct
T = [[A,            eye(m),     zeros(m,1), b];
      f',           zeros(1,m), 1,          0;
      zeros(1,n),   ones(1,m),  1,          0]
```

```
T = 4×9
    -1     0    -1     1    -1     1     0     0     1
     0    -1     2    -3    -2     0     1     0     2
     3     0     1     0     2     0     0     1     0
     0     0     0     0     0     1     1     1     0
```

```
for i = 1:m
    T(end,:) = T(end,:) - T(i,:);
end
T
```

```
T = 4×9
    -1     0    -1     1    -1     1     0     0     1
     0    -1     2    -3    -2     0     1     0     2
     3     0     1     0     2     0     0     1     0
     1     1    -1     2     3     0     0     1    -3
```

```
% Phase 1
min_col = 3; %1,1,-1,2,3,0,0,1
min_row = 3; %Ratios:inf,1,0

% Pivoting
new_T = zeros(size(T));
new_T(min_row,:) = T(min_row,:)/T(min_row,min_col);
for row = 1:size(T,1)
    if row ~= min_row
```

```
        new_T(row,:) = T(row,:) ...
            - T(row,min_col) * new_T(min_row,:);
    end
end
T = new_T
```

```
T = 4×9
     2     0     0     1     1     1     0     1     1
    -6    -1     0    -3    -6     0     1    -2     2
     3     0     1     0     2     0     0     1     0
     4     1     0     2     5     0     0     2    -3
```

```
% Optimal solution...
x3 = 0;
y1 = 1;
y2 = 2;

w = y1 + y2
```

```
w = 3
```

```
if w > 0
    disp('w > 0, infeasable')
end
```

```
w > 0, infeasable
```

```
% Confirmation:
disp('Confirmaiton with Linprog:')
```

```
Confirmaiton with Linprog
```

```
[~,~,exitflag]=linprog(f,[],[],C,d,zeros(n,1))
```

```
exitflag = -2
```

```
disp('exitflag = -2 => no feasable solution found')
```

```
exitflag = -2 => no feasable solution found
```

```
% Simplex Algorithm Function
function [x_min, f_min, n_iter, T] = simplex(f,A,b)
    %   Syntax  : [x_min, f_min, n_iter, T] = simplex(f,A,b)
    %
    %
    %   Purpose : Solves the problem
    %
    %               min      f'*x
    %               st.      A*x    <= b
    %                        x >= 0
    %   Assumes no equality constraints and x_i >= 0 forall i
    arguments
        f (:,1) double {mustBeNumeric,mustBeReal}
        A (:,:) double {mustBeNumeric,mustBeReal} = []
        b (:,1) double {mustBeNumeric,mustBeReal} = []
    end
    % Assuming everything inputed is good....
    max_iter = 20;
    % Setup
    n = size(f,1);
    num_s = size(A,1);% 0;

    T = [[A;f'],eye(size(A,1)+1),[b;0]];
    disp('Initial T = ');
    disp(T);

    n_iter = 0;
    ratios = zeros(size(T,1)-1,1);
    while any(T(end,1:end-1)<0) %Keeps going until optimal (final row
 >= 0)
        n_iter = n_iter + 1;
        % complicated way to find smallest value index
        min_val = min(T(end,1:end-1));
        [~, min_col] = find(T(end,1:end-1)==min_val,1,'first');
        % find pivot row
        for row = 1:(size(T,1)-1)
            if T(end,row) >= 0
                if T(row, min_col) > 0
                    ratios(row) = T(row,end) / T(row,min_col);
                else
                    ratios(row) = inf;
                end
            else
                ratios(row) = inf;
            end
        end
        min_val = min(ratios);
        [min_row,~] = find(ratios==min_val,1,'first');

        % Pivoting
        new_T = zeros(size(T));
        new_T(min_row,:) = T(min_row,:)/T(min_row,min_col);
```

```
        for row = 1:size(T,1)
            if row ~= min_row
                new_T(row,:) = T(row,:) ...
                    - T(row,min_col) * new_T(min_row,:);
            end
        end
        T = new_T;

        if n_iter >= max_iter
            error('too many iterations')
        end

    end
    disp('Final T = ');
    disp(T);


    % Calculating the basic variables
    j = 1;
    row = zeros(size(T,1),1);
    col = zeros(size(T,1),1);
    for i = 1:size(T,2)
        if nnz(T(:,i)) == 1
            col(j) = i;
            row(j) = find(T(:,i),1);
            j = j+1;
        end
    end

    % Solving for x values
    X = zeros([n,n+1]);
    for i = 1:n
        if col(i) <= n
            X(i,col(i)) = 1;
            X(i,end) = T(i,end);
        else
            X(i,:) = [A(i,:),b(i) - T(i,end)] ;
        end
    end
    X = rref(X);
    x_min = X(:,end);
    f_min = f'*x_min;
end

Initial T =
     2     1     1     1     0     0     0    15
    -1     2    -1     0     1     0     0     7
     1     5     5     0     0     1     0    25
     1     2    -7     0     0     0     1     0

Final T =
     2     1     1     1     0     0     0    15
     1     3     0     1     1     0     0    22
    -9     0     0    -5     0     1     0   -50
```

```
    15     9     0     7     0     0     1   105


x_min =

     0
     0
    15


f_min =

  -105


n_iter =

     1


T =

     2     1     1     1     0     0     0    15
     1     3     0     1     1     0     0    22
    -9     0     0    -5     0     1     0   -50
    15     9     0     7     0     0     1   105
```

Published with MATLAB® R2020b