LINEAR OPTIMIZATION PRACTICAL WORK RENDER

Télécom Physique Strasbourg. **Master 1 IRIV - Optimization 2022**

BERTRAND Simon

BFS : Basic Feasible Solution

Q1)

```
%Brute forcing the simplest BFS
function [costOpt, xOpt, BOpt] = LP_Bourrin(A,b,c)
    [m,n] = size(A);
    All_ind= nchoosek(1:n,m);
    solutions=zeros(n,size(All_ind,1));
    for i=1:size(All_ind,1)
        B= A(:,All_ind(i,:));
        x=zeros(1, n)';
        x(All_ind(i,:))=(B\b)';
        if sum(x>=0)== n
            solutions(:,i)= x; %Stack solutions into a matrix
        end
    end
    [costOpt, idxmin] = min(c'*solutions); %Compute min over all solutions
    xOpt=solutions(:,idxmin);
    BOpt=All_ind(idxmin,:);
end
```

We firstly generate all possible pairs of base indices and then compute the simplest BFS that is

$x = [xB\ 0]$ with the right base B.

We horizontal-stack all solutions to *[B^-1*b 0]* as columns of the "*solutions*" matrix. This h-stack is only realised when all solutions are superior or equal to zero.

Then we look for the minimum of the scalar products between all columns of the "*solutions*" matrix and the objective vector called "*c*". The cost function corresponds to the minimum value. We can then get the optimal solution *xOpt* corresponding to the base *BOpt*.

We found the optimal simplest BFS.

Q2)

```
function [cano_tab, costOpt, xOpt, BOpt] = LP_Simplex(A,b,c,v)
    % One phased method
    [m,n]=size(A);
    tableau = [A b ; c' 0];
    cano_tab=[eye(m) zeros(m,1) ; -tableau(end,v) 1]*[inv(tableau(1:end-1,v)) zeros(m,1); zeros(1,m) 1]*tableau;
    BOpt=[];
    while sum(cano_tab(end,:)>=0)~=size(cano_tab,2) %While solution is not BFS
        [~, q]=min(cano_tab(end,1:n)); %Compute min of last row
        BOpt(end+1)=q; %Stack the new base
        if sum(cano_tab(1:m,q)>0) < 1 %Check for full negative values
            disp("No solution : the problem is unbounded")
            break
        else
            yiq=cano_tab(1:m,q);
            yiq(yiq<0) = 0; %Remove negative elements
            [~, p]=min(cano_tab(1:m,end)./yiq); %Compute the min of column divided by column pivot
            pivot=cano_tab(p,q); %Get pivot value
            rowsInd=(1:m+1); rowsInd(p)=[]; %Prepare for rows operations
            cano_tab(p,:)=cano_tab(p,:)/pivot; %Normalize pivot row
            for row=rowsInd %For each row that is not the pivot's row
                cano_tab(row,:) = cano_tab(row,:) - cano_tab(row,q)*cano_tab(p,:);
            end
        end
    end
    xOpt=zeros(1,n-1);
    xOpt(BOpt)=cano_tab(1:end-1, end);
    costOpt=cano_tab(end,end);
end
```

LINEAR OPTIMIZATION PRACTICAL WORK RENDER

Télécom Physique Strasbourg. **Master 1 IRIV - Optimization 2022**

BERTRAND Simon

We firstly create the tableau to transform it into the canonical tableau using 2 times the matrix dot product with 3 different built matrices with the base corresponding to *v*. We use the matrix form of the simplex algorithm. Once created, we verify if the current last row is a BFS and if not, we proceed to the while. During this while, we find the q index corresponding to the column index of the minimum last row values. If there are no positive values in the q-column, we break the while because the problem is unbounded. If not, we compute p, the argmin of the last column divided by the q-column such as the values that are strictly negative in the q-column are replaced by zero to be ignored by the computation of the minimum. The pivot is the p,q element of the canonical tableau. We divide the row corresponding to the pivot by the scalar pivot value and then, we annulate the others q-columns coefficient doing rows operations with the pivot row. We then go back to the condition of the while to check if the current solution is a BFS or not. Once the BFS is found, we are able to return the *xOpt* with its corresponding base *BOpt* and also the optimal cost.

We have implemented the one phased method.

Q3)

```
function [costOpt, xOpt, BOpt] = LP_Two_Phase_Simplex(A,b,c)
    % Phase 1 : resolving a virtual problem to get the
    %initial basic feasible solution
    [m,n]=size(A);
    B_virtual=n+1:n+m; % Identical matrix base
    init_A = [A eye(m)];% Concatenate identical matrix
    init_c=[zeros(n,1);ones(m,1)]; % Define virtual objective function
    [cano_tab, ~, ~, B_init] = LP_Simplex(init_A,b,init_c,B_virtual);

    % Phase 2 : Return to the unvirtual function
    cano_tab(:,B_virtual)=[]; %Delete virtual columns
    tableau=[cano_tab(1:end-1,:) ; c' 0]; % Create tableau
    cano_tab_phase2=[eye(m) zeros(m,1) ; -tableau(end,B_init) 1]*[inv(tableau(1:end-1,B_init)) zeros(m,1); zeros(1,m) 1]*tableau;
    %Created the canonical tableau
    xOpt=zeros(1,n);
    %Push solutions with the right base
    xOpt(B_init)=cano_tab_phase2(1:end-1, end);
    costOpt=xOpt*c;
    BOpt=B_init;
end
```

The two phased method is used when a base *v* cannot be found directly. To proceed with the method, we firstly resolve a virtual problem that is based on a column's extensive with the identity matrix columns. The *c* coefficient is made of zeros for the values corresponding to the real variables and of ones for the values corresponding to the virtual variables. This problem can be resolved with the Q2 Matlab function. The basis *v* used corresponds to the virtual variables added because it is the basis for a simple BFS.

We finish the phase 1 with the canonical tableau and an initial basis *B_init* corresponding to a BFS.

For the second phase, we delete the virtual columns and create a tableau to transform it into a canonical tableau. Then we can get the costOpt, xOpt, and BOpt.

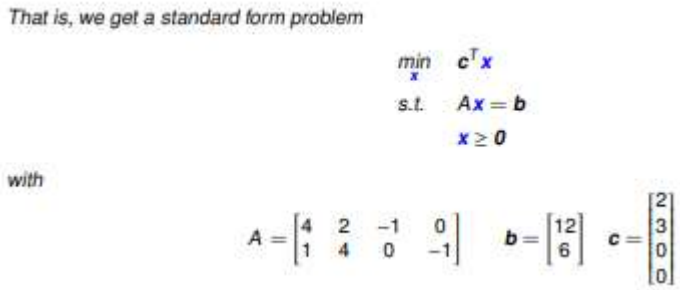
LINEAR OPTIMIZATION PRACTICAL WORK RENDER

Télécom Physique Strasbourg. **Master 1 IRIV - Optimization 2022**

BERTRAND Simon

Verification:

We will use the subject of p-45 of the Linear Programming PDF. The subject is :

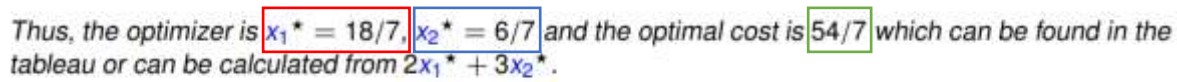



```
%RESOLVE THE FOLLOWING PROBLEM :
% min of f:(x1,x2)->2*x1 + 3*x2
% s.t. :
% 4x1 + 2x2 − x3 = 12
% x1 + 4x2 − x4 = 6
% x1 ≥ 0, . . . , x4 ≥ 0

A=[4 2 -1 0;1 4 0 -1];
b=[12;6];
c=[2;3;0;0];

%Two phase if a basic feasible solution is not easy to find
[costOpt, xOpt, BOpt] = LP_Two_Phase_Simplex(A,b,c)
```

We get as return:

```
costOpt =

    7.7143


xOpt =

    2.5714    0.8571         0         0


BOpt =

     2     1
```

Where solutions are:




```
>> [18/7, 6/7, 54/7]

ans =

    2.5714    0.8571    7.7143
```