
```

%Leidy Vargas M
%15/05/2019
%Enunciado1 Maximización
%Un estanque de peces es abastecido cada primavera con dos especies:
    beta y globo;
% si hay dos tipos de comida f1 y f2 disponibles en el tanque. El peso
    promedio de los peces y
% el requerimiento promedio de alimento para cada pez; esta dado en la
    siguiente tabla:
%Especie/ F1/ F2 / Peso promedio
% beta    2    3    3 lb
% globo   3    1    2 lb
%Si existen 600 lb de comida f1 y 300 lb de comida f2 diariamente.
% ¿Cuántos peces deben existir en la pecera; dado que lo mínimo para
    lo cual fue construida es de 400 lb?
%X1:No peces beta que deben haber X2:No peces globo que deben haber

                                %Max z=x1+x2
                                %s.a:3x1+2x2>=400
                                %    2x1+3x2<=600
                                %    3x1+x2<=300
                                %    x1, x2 #0

%A=[1 1]; %Se define A como funcion objetivo

%B=[ 3 2; %%Se define B como restricciones
    % 2 3;
    % 3 1];

%C=[ 400; %% vector de recursos de las desigualdades
    % 600;
    % 300];

function granM_Aplicacion1()
clc;
clear all;
A=[1 1];
B=[ 3 2;
    2 3;
    3 1];
C=[ 400; %% vector de recursos de las desigualdades
    600;
    300];
% Ventana para definir la función objetivo
prompt = {'Función objetivo = ', 'max == 1 or min==2', 'Numero de
    restricciones = '};
lineno = 1;
title = 'Ingreso de Datos';
def = {'A', '1', '3'};
options.Resize = 'on';
a = inputdlg(prompt,title,lineno,def,options);
a = char(a);
[m,n] = size(a);

```

```

    if n(1)~= '<' && isempty(sc2)
        sc2=zeros(nbr,1);
    end
    switch str1(1,i).Type
        case '<='
            v_e=[v_e second(i)];
            sc1(j,length(v_e))=1;
            v_b=[v_b,second(i)];

            case '>='
                v_e=[v_e 0];
                sc1(j,length(v_e))=-1;
                v_ari=[v_ari second(i)];
                sc2(j,length(v_ari))=1;
                v_b=[v_b,second(i)];

            case '='
                v_ari=[v_ari second(i)];
                sc2(j,length(v_ari))=1;
                sc1(j,length(v_ari))=0;
                v_b=[v_b,second(i)];

    end
    j=j+1;
end
%=====
sc      =[sc,sc1,sc2]; %Nueva Matriz de restricciones con variables
        artificiales y de holgura añadidas
vari     =[];
vari_a   =[];
vari_e   =[];
vari_ar  =[];
for i=1:size(sc,2)
    str1(1,i).vari=['x',num2str(i)];
    vari=[vari,str1(1,i).vari,' '];
    if i<length(v_a)
        vari_a=[vari_a,str1(1,i).vari,' '];
    elseif i<=length(v_a)+length(v_e)
        vari_e=[vari_e,str1(1,i).vari,' '];
    else
        vari_ar=[vari_ar,str1(1,i).vari,' '];
    end
end
%Primera iteración
x=[v_a,v_e,v_ari];
if ~isempty(v_ari~=0)
    v_ar     = ones(1,length(v_ari));
    if type==1
        v_ar=-M*length(v_ari).*v_ar;
    else
        v_ar=M*length(v_ari).*v_ar;
    end
else v_ar=[];
end

```

```

Cj=[cout,0.*v_e,v_ar];
Vb=[];
Q=v_b;
Ci=[];
tabl=[];
for i=1:length(Q)
    tabl=[tabl; ' | '];
    str2(1,i).valeur=Q(i);
    ind=find(x==Q(i));
    str2(1,i).var_base=str1(1,ind).vari;
    Vb=[Vb,str2(1,i).var_base, ' '];
    Ci=[Ci,Cj(ind)];
end
Z=sum(Ci.*Q);
for i=1:length(Cj)
    Zj(i)=sum(Ci'.*sc(:,i));
end
Cj_Zj=Cj-Zj;
l=[];
for i=1:nbr
    if length(str2(1,i).var_base)==2
        l=[l;str2(1,i).var_base, ' '];
    else
        l=[l;str2(1,i).var_base];
    end
end
fprintf('\n');
disp('===== Problema en forma estandar
=====');
disp(['Variables : ',vari]);
disp(['          -Variables No Básicas      : ',vari_a]);
disp(['          -Variables Básicas         : ',vari_e]);
disp(['          -Variables Artificiales    : ',vari_ar]);
disp('=====');
disp(' ');
disp('===== Tabla 0
=====');
disp(['Inicialización de variables : ',vari]);
disp(['          -Variables No Básicas      :
',num2str(v_a)]);
disp(['          -Variables Básicas         :
',num2str(v_e)]);
disp(['          -Variables Artificiales    :
',num2str(v_ar)]);
disp('=====');
disp(' ');
disp(['Cj          : ',num2str(Cj)]);
disp('_____');
disp([tabl,num2str(Ci'),tabl,l,tabl,num2str(Q'),tabl,num2str(sc),tabl]);
disp('_____');
disp(['Zj          : ',num2str(Zj)]);
disp(['Cj-Zj       : ',num2str(Cj-Zj)]);
disp(['Z          : ',num2str(Z)]);
disp('_____');

```

```

disp(' ');
%Iteraciones de Simplex Gran M
t      = 1;
arret  = 1;
while arret==1
    if type==1
        num=max(Cj_Zj);num=num(1);
        num1=find(Cj_Zj==num);num1=num1(1);
        V_ent=str1(1,num1).vari;
    else
        g=min(Cj_Zj);g=g(1);
        num1=find(Cj_Zj==g);num1=num1(1);
        V_ent=str1(1,num1).vari;                ['x',num2str(num1)];
    end
    b=sc(:,num1);
    k=0;d=10000;
    for i=1:length(Q)
        if b(i)>0
            div=Q(i)/b(i);
            if d>div
                d=div;
                k=i;
            end
        end
    end
    if k~=0
        num2=k;
    else
        disp('No se puede encontrar solución : La solución es
infactible ');
        break;
    end
    V_sort=str2(1,num2).var_base;
    str2(1,num2).var_base=str1(1,num1).vari;
    pivot=sc(num2,num1);
    Ci(num2)=Cj(num1);
    sc(num2,:)=sc(num2,:)./pivot;
    Q(num2)=Q(num2)/pivot;
    h=size(sc,1);
    for i=1:h
        if i~=num2
            Q(i)=Q(i)-sc(i,num1)*Q(num2);
            sc(i,:)=sc(i,:)-sc(i,num1).*sc(num2,:);
        end
    end
    Z=sum(Ci.*Q);
    for i=1:size(sc,2)
        Zj(i)=sum(Ci'.*sc(:,i));
    end
    Cj_Zj=Cj-Zj;
    l=[];V=[];
    for i=1:nbr
        if length(str2(1,i).var_base)==2

```

```

        l=[1;str2(1,i).var_base, ' '];
        V=[V,l(i,:), ' '];
    else
        l=[1;str2(1,i).var_base];
        V=[V,l(i,:), ' '];
    end
end
Vb      = V;
disp(['===== Tabla ',num2str(t),'
=====']);
disp(['Variable de entrada : ',num2str(V_ent)]);
disp(['Variable de salida : ',num2str(V_sort)]);
disp(['Pivote : ',num2str(pivot)]);
disp(['Variables Básicas : ',num2str(Vb)]);

disp('=====');
disp(' ');
disp(['Cj : ',num2str(Cj)]);

disp('_____');

disp([tabl,num2str(Ci'),tabl,1,tabl,num2str(Q'),tabl,num2str(sc),tabl]);

disp('_____');
disp(['Zj : ',num2str(Zj)]);
disp(['Cj-Zj : ',num2str(Cj-Zj)]);
disp(['Z : ',num2str(Z)]);

disp('_____');
disp(' ');
disp(' ');
t      = t+1;
if type==1
    a=max(Cj_Zj);a=a(1);
    if a<=0
        break;
    end
else
    a = min(Cj_Zj);a=a(1);
    if a>=0 break;
    end
end
end
p      = num2str(Z);
disp(['Resultado F.O. OPTIMO : ',num2str(Z)]);
disp('=====');
disp('SOLUCIÓN')
disp(['Los peces tipo beta deberán ser: ',num2str(Q(3)),' peces']);
disp(['Los peces tipo globo deberán ser: ',num2str(Q(1)),' peces']);
disp(['Y la cantidad que debe existir en la pecera es
de:',num2str(Z),' peces']);
%k      = msgbox( p,'RESULTADO F.O. OPTIMO :')

```

===== Problema en forma estandar

=====

Variables : x1 x2 x3 x4 x5 x6

-Variables No Básicas : x1

-Variables Básicas : x2 x3 x4 x5

-Variables Artificiales : x6

=====

===== Tabla 0

=====

Inicialización de variables : x1 x2 x3 x4 x5 x6

-Variables No Básicas : 0 0

-Variables Básicas : 0 600 300

-Variables Artificiales : -3000

=====

Cj : 1 1 0 0 0 -3000

	-3000		x6		400		3	2	-1	0	0	1	
	0		x4		600		2	3	0	1	0	0	
	0		x5		300		3	1	0	0	1	0	

Zj : -9000 -6000 3000 0 0 -3000

Cj-Zj : 9001 6001 -3000 0 0 0

Z : -1200000

=====

===== Tabla 1

=====

Variable de entrada : x1

Variable de salida : x5

Pivote : 3

Variables Básicas : x6 x4 x1

=====

Cj : 1 1 0 0 0 -3000

	-3000		x6		100		0			1		-1		0
-1			1											
	0		x4		400		0			2.3333		0		1
-0.66667							0							
	1		x1		100		1			0.33333		0		0
0.33333							0							

Zj : 1 -2999.6667 3000 0

3000.3333 -3000

Cj-Zj : 0 3000.6667 -3000 0

-3000.3333 0

Z : -299900

=====

===== Tabla 2

=====

Variable de entrada : x2
Variable de salida : x6
Pivote : 1
Variables Básicas : x2 x4 x1

Cj	: 1	1	0	0	0	-3000
1 x2	100 0	1	-1	0		
-1	1					
0 x4	166.6667 0	0	2.3333	1		
1.6667	-2.3333					
1 x1	66.66667 1	0	0.33333	0		
0.66667	-0.33333					
Zj	: 1	1	-0.66667	0		
-0.33333	0.66667					
Cj-Zj	: 0	0	0.66666667	0		
0.33333333	-3000.6667					
Z	: 166.6667					

===== Tabla 3

=====

Variable de entrada : x3
Variable de salida : x4
Pivote : 2.3333
Variables Básicas : x2 x3 x1

Cj	: 1	1	0	0	0	-3000
1 x2	171.4286 0	1	0	0.42857		
-0.28571	0					
0 x3	71.42857 0	0	1	0.42857		
0.71429	-1					
1 x1	42.85714 1	0	0	-0.14286		
0.42857	0					
Zj	: 1	1	0	0.28571		
0.14286	0					
Cj-Zj	: 0	0	0	-0.28571429		
-0.14285714	-3000					
Z	: 214.2857					

Resultado F.O. OPTIMO : 214.2857

=====

SOLUCIÓN

Los peces tipo beta deberán ser: 42.8571 peces
Los peces tipo globo deberán ser: 171.4286 peces
Y la cantidad que debe existir en la pecera es de:214.2857 peces

Published with MATLAB® R2019a