
```

%Leidy Vargas M
%15/05/2019
%Enunciado2
%Un artesano fabrica trenes y camiones de juguetes utilizando
    tornillos,
% bloques y ruedas como componentes, en la semana próxima dispone de
% 8000 6000 y 6300 componentes respectivamente, los beneficios por
    tren
% y camión son 1.6 euros/unidad y 1.4 euros/unidad
% Producto/ Tornillos/ Bloques / Ruedas/
% X1 tren          10      15      18
% X2 camión        20      10      6
                                %Max z= 1,6 X1+1,4 X2
                                %s.a: 10X1+20X2<=8000
                                %      15X1+10X2<=6000
                                %      18X1+6X2<=6300
                                %      x1, x2 #0
%A=[1.6 1.4]; %Se define A como funcion objetivo

%B=[ 10 20; %%Se define B como restricciones
    15 10;
    18 6];

%C=[ 8000; %% vector de recursos de las desigualdades
    6000;
    6300];

function simplexAplicacion2()
clc;
clear all;
A=[1.6 1.4];

B=[ 10 20;
    15 10;
    18 6];

C=[ 8000;
    6000;
    6300];

% Ventana para definir la función objetivo
prompt = {'Función objetivo = ', 'max == 1 ', '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);
cout = eval(a(1,:)); %Se transforman los valores (cadena de
    caracteres) ingresados en la FO a un vector de enteros

```

```

type      = eval(a(2,1)); %Se transforman los valores (cadena de
                        caracteres) ingresados en el tipo de FO a un entero
nbr        = eval(a(3,1)); %Se transforman los valores (cadena de
                        caracteres) ingresados en el numero de restricciones a un entero
str1       = struct('vari',{},'Type',{});
str2       = struct('var_base',{},'valeur',{});
%=====
% Ventana para definir restricciones
for i=1:nbr %Se definen los tipos de restricciones en orden
    prompt  = {strcat('Ingrese el tipo de restricción para la
condición ',num2str(i),' (<=,>=,=):')};
    title   = 'Ingreso, de Datos';
    def     = {'<='};
    options.Resize = 'on';
    p = inputdlg(prompt,title,lineno,def,options);
    p = char(p);
    opert = p;
    str1(1,i).Type = opert;
end
%=====
% Ventana para definir coeficientes de las restricciones
prompt  = {'Ingrese la matriz de restricciones'};
lineno  = 1;
title   = 'Ingreso de Datos';
def     = {'B'};
options.Resize='on';
t       = inputdlg(prompt,title,lineno,def,options);
t       = char(t);
sc      = eval(t); %Se transforman los valores (cadena de caracteres)
                        ingresados en el campo, a un matriz de enteros
%=====
% Ventana para definir coeficientes de 'b'
prompt  = {'Ingrese el vector de valores independientes b'};
lineno  = 1;
title   = 'Ingreso de Datos';
def     = {'C'};
options.Resize = 'on';
u       = inputdlg(prompt,title,lineno,def,options);
u       = char(u);
second  = eval(u);
%=====

sc1      = []; %Matriz de Variables de holgura
v_a      = zeros(1,length(cout));
v_e      = [];
v_b      = [];
j        = 1;
%Paso a forma estandar
for i=1:nbr
    n = str1(1,i).Type;
    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)];

        end
        j=j+1;
    end
    %=====
    sc      =[sc,sc1]; %Nueva Matriz de restricciones con variables de
    %holgura añadidas
    vari     =[];
    vari_a   =[];
    vari_e   =[];

    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,' '];
        end
    end
    %Primera iteración
    x=[v_a,v_e];
    Cj=[cout,0.*v_e];
    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
end

```

```

fprintf('\n');
disp('===== Problema en forma estandar
=====');
disp(['Variables : ',vari]);
disp(['          -Variables No Básicas      : ',vari_a]);
disp(['          -Variables Básicas         : ',vari_e]);
disp('=====');
disp(' ');
disp('===== Tabla 0
=====');
disp(['Inicialización de variables : ',vari]);
disp(['          -Variables No Básicas      :
',num2str(v_a)]);
disp(['          -Variables Básicas         :
',num2str(v_e)]);
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(' ');
%Iteraciones de Simplex
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=[l;str2(1,i).var_base, ' '];
            V=[V,l(i,:), ' '];
        else
            l=[l;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(['Se deben producir ',num2str(Q(3)),' trenes ']);
disp(['Se deben producir ',num2str(Q(2)),' camiones ']);
disp(['Obteniendo un beneficio de:',num2str(Z),' dólares']);
%k      = msgbox( p,'RESULTADO F.O. OPTIMO :')

```

===== Problema en forma estandar

=====

Variables : x1 x2 x3 x4 x5

-Variables No Básicas : x1

-Variables Básicas : x2 x3 x4 x5

=====

===== Tabla 0

=====

Inicialización de variables : x1 x2 x3 x4 x5

-Variables No Básicas : 0 0

-Variables Básicas : 8000 6000 6300

=====

Cj : 1.6 1.4 0 0
0

/ 0 /	x3	/ 8000 /	10	20	1	0	0	/
/ 0 /	x4	/ 6000 /	15	10	0	1	0	/
/ 0 /	x5	/ 6300 /	18	6	0	0	1	/

Zj : 0 0 0 0 0

Cj-Zj : 1.6 1.4 0 0
0

Z : 0

=====

===== Tabla 1				
=====				
Variable de entrada : x1				
Variable de salida : x5				
Pivote : 18				
Variables Básicas : x3 x4 x1				
=====				
Cj	: 1.6	1.4	0	0
0				
0 x3 4500 0 16.6667 1 0				
-0.555556				
0 x4 750 0 5 0 1				
-0.833333				
1.6 x1 350 1 0.333333 0 0				
0.055556				
Zj	: 1.6	0.53333	0	0
0.088889				
Cj-Zj	: 0	0.86667	0	0
-0.088889				
Z	: 560			

===== Tabla 2				
=====				
Variable de entrada : x2				
Variable de salida : x4				
Pivote : 5				
Variables Básicas : x3 x2 x1				
=====				
Cj	: 1.6	1.4	0	0
0				
0 x3 2000 0 0 1 -3.3333				
2.2222				
1.4 x2 150 0 1 0 0.2				
-0.16667				
1.6 x1 300 1 0 0 -0.066667				
0.11111				
Zj	: 1.6	1.4	0	0.17333
-0.055556				
Cj-Zj	: 0	0	0	-0.17333
0.055556				
Z	: 690			

===== Tabla 3				
=====				

Variable de entrada : x5
 Variable de salida : x3
 Pivote : 2.2222
 Variables Básicas : x5 x2 x1

Cj									
0									
<hr/>									
/	0	/	x5	/	900	/	0		
/							0	0.45	-1.5
/	1.4	/	x2	/	300	/	0		
/							1	0.075	-0.05
/	1.6	/	x1	/	200	/	1		
/							0	-0.05	0.1
<hr/>									
Zj									
0									
Cj-Zj									
0									
Z									
<hr/>									

Resultado F.O. OPTIMO : 740

SOLUCIÓN

Se deben producir 200 trenes
 Se deben producir 300 camiones
 Obteniendo un beneficio de:740 dólares

Published with MATLAB® R2019a