
```

%Leidy Vargas M
%15/05/2019
%Enunciado1
%Sobre dos alimentos diferentes tenemos la siguiente información por
    kilogramo:
%Alimento/ Calorías/ Proteínas (gr) / /Precio (ptas)
%   A           1000           25                60
%   B           2000           100               210
%Hallar el coste mínimo de una dieta formada sólo por este tipo de
    alimentos y que
%al menos aporte 3000 calorías y 100 gramos de proteínas.
%x1 = kilogramos de alimento A.
%x2 = kilogramos de alimento B.
                                %min z= = 60x1 + 210x2
                                %s.a.: 1000 x1+ 2000x2 #3000
                                %      25x1 + 100x2 #100
                                %              x1, x2 #0
%A=[60 210]; %Se define A como funcion objetivo

%B=[1000 2000; %%Se define B como restricciones
% 25 100];

%C=[ 3000; %% vector de recursos de las desigualdades
% 100];

function Aplicacion1()
clc;
clear all;
A=[1000 2000 3000;
    25 100 100;
    60 210 0]; %Se define A como matriz inicial
TRANSPUESTA = A.'
A=TRANSPUESTA([3],[1,2])
B=TRANSPUESTA([1,2],[1,2])
C=TRANSPUESTA([1,2],[3])

% 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','2'};
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     = {'[1000 25;2000 100]'};
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 una matriz de enteros
%=====
% Ventana para definir coeficientes de 'b'
prompt = {'Ingrese el vector de valores independientes b'};
lineno = 1;
title   = 'Ingreso de Datos';
def     = {'[60;210]'};
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   =[];
    vari_ar  =[];
    for i=1:size(sc,2)
        str1(1,i).vari=['y',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
    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('=====');
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 ');
    end
end

```

```

        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
disp(['Resultado F.O. OPTIMO : ',num2str(Z)]);
disp('=====');
disp('SOLUCIÓN')
disp(['Siendo : ',num2str(Cj_Zj(3)*-1),' kilogramos de alimento A.']);
disp(['Siendo: ',num2str(Cj_Zj(4)*-1),' kilogramos de alimento B.']);
disp(['El coste mínimo de una dieta formada sólo por este tipo de
    alimentos es:',num2str(Z),' pesetas']);

%k  = msgbox( p,'RESULTADO F.O. OPTIMO :')

```

TRANSPUESTA =

1000	25	60
2000	100	210
3000	100	0

A =

3000	100
------	-----

B =

1000	25
2000	100

C =

60
210

===== Problema en forma estandar

=====

Variables : y1 y2 y3 y4

-Variables No Básicas : y1

-Variables Básicas : y2 y3 y4

=====

===== Tabla 0

=====

Inicialización de variables : y1 y2 y3 y4

-Variables No Básicas : 0 0

-Variables Básicas : 60 210

=====

Cj : 3000 100 0 0

0 y3 60 1000	25	1	0
0 y4 210 2000	100	0	1

Zj : 0 0 0 0

Cj-Zj : 3000 100 0 0

Z : 0

===== Tabla 1

=====

Variable de entrada : y1

Variable de salida : y3

Pivote : 1000

Variables Básicas : y1 y4

=====

Cj : 3000 100 0 0

3000 y1 0.06 1	0.025	0.001	0
0 y4 90 0	50	-2	1

Zj : 3000 75 3 0

Cj-Zj : 0 25 -3 0

Z : 180

===== Tabla 2

=====

Variable de entrada : y2

Variable de salida : y4

Pivote : 50

Variables Básicas : y1 y2

=====

Cj : 3000 100 0 0

3000 y1 0.015 1	0	0.002	-0.0005
100 y2 1.8 0	1	-0.04	0.02

Zj : 3000 100 2

0.5

Cj-Zj : 0 0 -2 -0.5

Z : 225

Resultado F.O. OPTIMO : 225

=====

SOLUCIÓN

Siendo : 2 kilogramos de alimento A.

Siendo: 0.5 kilogramos de alimento B.

*El coste mínimo de una dieta formada sólo por este tipo de alimentos
es:225 pesetas*

Published with MATLAB® R2019a