MAT2003

LAB – 7

SUBMITTED BY : SRIHARSHITHA DEEPALA

REG NO : 19BCD7246

SLOT : L6

QUESTION – 1:(VAM METHOD)

	D1	D2	D3	D4	SUPPLIES
S1	6	4	1	5	14
S2	8	9	2	7	16
S3	4	3	6	2	5
REQUIRED	6	10	15	4	35

CODE – 1:

```
function [ z,x ] = vogel(s,d,c,m,n );
s = input("Enter the supplies of column vector :");
d = input("Enter the demands of row vector");
c = input("Enter the cost matrix :");
m = input("Enter the value of m");
n = input("Enter the value of n");
x=zeros(m,n);
numbasic=n+m-1;
for j=1:n
```

```matlab
for i=1:m
c1(i,j)=c(i,j);
end
end
for i=1:m
s1(i)=s(i);
end
for j=1:n
d1(j)=d(j);
end
iteration=0;
for k=1:n+m-1
iteration=iteration+1;
%% Row Difference
minrow1=zeros(m,1);
minrow2=zeros(m,1);
jmin=zeros(1,m);


for i=1:m
min1=inf;
for j=1:n
if c1(i,j)<min1
min1=c1(i,j);
jmin(i)=j;% position of min1 on each row
end
end
minrow1(i)=min1;
end;
for i=1:m
min2=inf;
for j=1:n
if j~=jmin(i)
if c1(i,j)<=min2
min2=c1(i,j);
end
end
end
minrow2(i)=min2;
end
%% Column Difference
mincol1=zeros(1,n);
```

```
mincol2=zeros(1,n);
imin=zeros(n,1);
for j=1:n
    minR1=inf;
    for i=1:m
        if c1(i,j)<minR1
            minR1=c1(i,j);
            imin(j)=i;
        end
    end
    mincol1(j)=minR1;
end
for j=1:n
    minR2=inf;
    for i=1:m
      if i~=imin(j)
          if c1(i,j)<=minR2
              minR2=c1(i,j);
          end
      end
    end
    mincol2(j)=minR2;
end
%% Difference
diffrow=zeros(m,1);
diffcol=zeros(1,n);
for i=1:m
    diffrow(i)=minrow2(i)-minrow1(i);
end
for j=1:n
    diffcol(j)=mincol2(j)-mincol1(j);
end
%% The greatest difference
    R=0;
    Row=zeros(m,1);
    for i=1:m
        if diffrow(i)>=R
            R=diffrow(i);
            iminrow=i;
        end
    end
    Row(iminrow)=R;
    S=0;
    Col=zeros(1,n);
    for j=1:n
        if diffcol(j)>=S
            S=diffcol(j);
            jmincol=j;
        end
    end
    Col(jmincol)=S;
    great=zeros(1,n);
    for j=1:n
    if S>=R
        great(jmincol)=Col(jmincol);
        Colline=1;
    else
        great(iminrow)=Row(iminrow);
        Colline=0;
    end
    end

%% Search the entry cell
if Colline==1
j=jmincol;
R1=inf;
for i=1:m
if c1(i,jmincol)<=R1
R1=c1(i,jmincol);
igreat=i;
end
end
if s1(igreat)>d1(jmincol)
x(igreat,jmincol)=d1(jmincol);
s1(igreat)=s1(igreat)-d1(jmincol);
d1(jmincol)=0;
eliminaterow=0;
elseif s1(igreat)<d1(jmincol)
x(igreat,jmincol)=s1(igreat);
d1(jmincol)=d1(jmincol)-s1(igreat);
s1(igreat)=0;
eliminaterow=1;
elseif s1(igreat)==d1(jmincol)
x(igreat,jmincol)=s1(igreat);
```

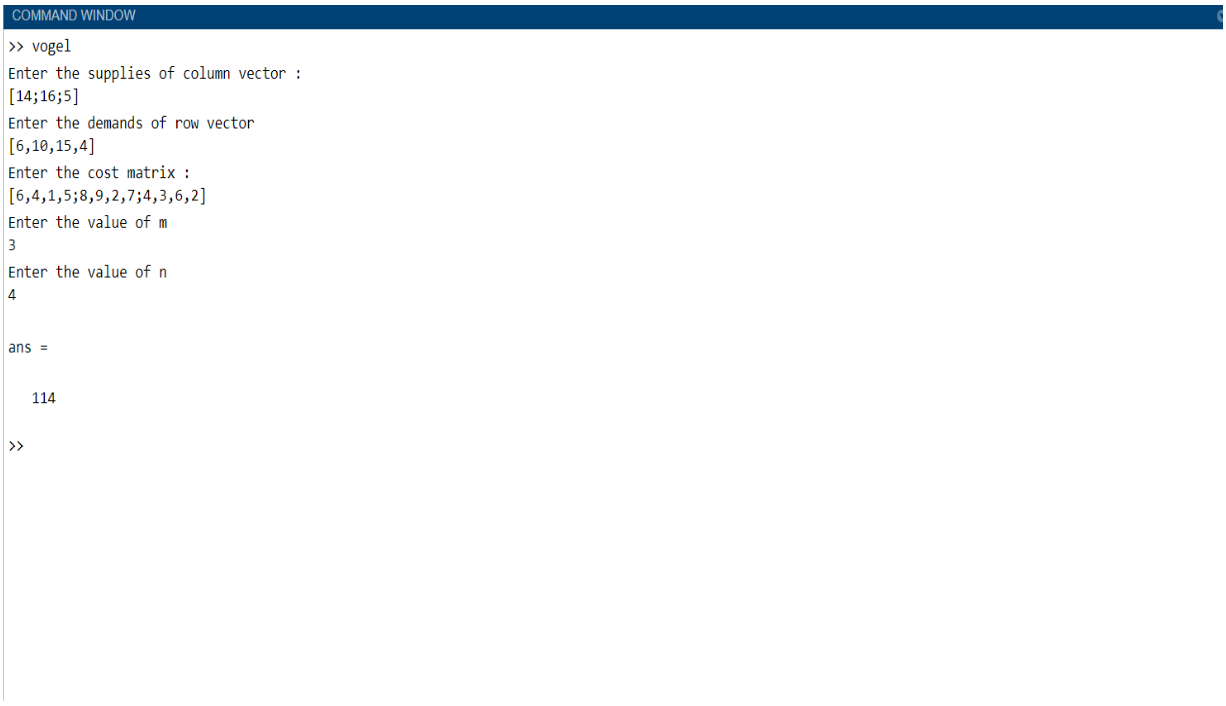
```
d1(jmincol)=0;
s1(igreat)=0;
eliminaterow=2;
end
% Eliminate a column or a row
if eliminaterow==0
for i=1:m
c1(i,jmincol)=inf;
end
elseif eliminaterow==1
for j=1:n
c1(igreat,j)=inf;
end
elseif eliminaterow==2
for i=1:m
c1(i,jmincol)=inf;
end
for j=1:n
c1(igreat,j)=inf;
end
end
else % Colline=0;
i=iminrow;
R2=inf;
for j=1:n
if c1(iminrow,j)<R2
R2=c1(iminrow,j);
jgreat=j;
end
end
if s1(iminrow)>d1(jgreat)
x(iminrow,jgreat)=d1(jgreat);
s1(iminrow)=s1(iminrow)-d1(jgreat);
d1(jgreat)=0;
eliminaterow=0;
elseif s1(iminrow)<d1(jgreat)
x(iminrow,jgreat)=s1(iminrow);
d1(jgreat)=d1(jgreat)-s1(iminrow);
s1(iminrow)=0;
eliminaterow=1;
elseif s1(iminrow)==d1(jgreat)
x(iminrow,jgreat)=s1(iminrow);
```

```
d1(jgreat)=0;
s1(iminrow)=0;
eliminaterow=2;
end
% Eliminate a column or a row
if eliminaterow==0
for i=1:m
c1(i,jgreat)=inf;
end
elseif eliminaterow==1
for j=1:n
c1(iminrow,j)=inf ;
end
elseif eliminaterow==2
for i=1:m
c1(i,jgreat)=inf
end
for j=1:n
c1(iminrow,j)=inf;
end
end
end
%% Calculate the objective function
z=0;
for j=1:n
for i=1:m
if x(i,j)>0
z=z+c(i,j)*x(i,j);
end
end
end
sums=0;
for i=1:m
sums=sums+s1(i);
end
sumd=0;
for j=1:n
sumd=sumd+d1(j);
end
if (sums & sumd) ==0
return
end
```

```
end
%% The degeneracy
countx=0;
for i=1:m
for j=1:n
if x(i,j)>0
countx=countx+1;
x1(i,j)=x(i,j);
x2(i,j)=x(i,j);
end
end
end
if countx>=numbasic
disp('Total cost of non-degeneracy VAM');
disp(z);
else
disp('Total cost of degeneracy VAM');
disp(z);
end


end
```

OUTPUT:



```
>> vogel
Enter the supplies of column vector :
[14;16;5]
Enter the demands of row vector
[6,10,15,4]
Enter the cost matrix :
[6,4,1,5;8,9,2,7;4,3,6,2]
Enter the value of m
3
Enter the value of n
4

ans =

   114

>>
```

CODE - 2

```
function [ Solution, OverallCost ] = VAM(Costs, supply,
demands)
Costs=input("Enter the cost matrix :");
supply=input("Enter the vector of supplies :");
demands=input("Enter the vector of demands :");
C_start = Costs;
C = C_start;
m = size(C,1);
n = size(C,2);
a = supply;
b = demands;
X = zeros(m,n);
stop = 0;
while stop == 0
  for i = 1:m
     for j = 1:n
        if a(i,1) == 0
          C(i,j) = max(C(:,j));
        end
        if b(1,j) == 0
          C(i,j) = max(C(i,:));
        end
     end
  end
  C_sort_col = sort(C,1);
  C_sort_row = sort(C,2);
  Diff_customer = abs(C_sort_col(1,:) - C_sort_col(2,:));
  Diff_supplier = abs(C_sort_row(:,1) - C_sort_row(:,2));
  for i = 1:m
    if a(i,1) == 0
      Diff_supplier(i,1) = 0;
    end
  end
  for j = 1:n
     if b(1,j) == 0
     Diff_customer(1,j) = 0;
     end
  end
  Max_Diff_customer = max(Diff_customer);
  Max_Diff_supplier = max(Diff_supplier);
  Customer_nr =
find(Diff_customer==max(Max_Diff_customer,Max_Diff_supplie
r));
  Supplier_nr =
find(Diff_supplier==max(Max_Diff_customer,Max_Diff_supplie
r));
  if isempty(Customer_nr) == 0
     Supplier_nr_ = find(C(:,Customer_nr(1)) ==
min(C(:,Customer_nr(1))));
     X(Supplier_nr_(1),Customer_nr(1)) =
min(a(Supplier_nr_(1),1),b(1,Customer_nr(1)));
     a(Supplier_nr_(1),1) = a(Supplier_nr_(1),1) -
X(Supplier_nr_(1),Customer_nr(1));
     b(1,Customer_nr(1)) = b(1,Customer_nr(1)) -
X(Supplier_nr_(1),Customer_nr(1));
     Supplier_nr = [];
  end
  if isempty(Supplier_nr) == 0
     Customer_nr_ = find(C(Supplier_nr(1),:) ==
min(C(Supplier_nr(1),:)));
     X(Supplier_nr(1),Customer_nr_(1)) =
min(a(Supplier_nr(1),1),b(1,Customer_nr_(1)));
     a(Supplier_nr(1),1) = a(Supplier_nr(1),1) -
X(Supplier_nr(1),Customer_nr_(1));
     b(1,Customer_nr_(1)) = b(1,Customer_nr_(1)) -
X(Supplier_nr(1),Customer_nr_(1));
  end
%Stop condition:
  a1 = a > 0;
  b1 = b > 0;
  if sum(a1) == 1
     stop = 1;
    for j = 1:n
        if b(j) > 0
          X(a1 == 1,j) = b(j)
        end
    end
  end
  if sum(b1) == 1
    stop = 1;
      for i = 1:m
         if a(i) > 0
```

```
            X(i,b1 == 1) = a(i)
         end
      end
   end
end
Solution = X
OverallCost = sum(sum(C_start .* X))
end
```

OUTPUT:

```
VAM
Enter the cost matrix :
[6,4,1,5;8,9,2,7;4,3,6,2]
Enter the vector of supplies :
[14;16;5]
Enter the vector of demands :
[6,10,15,4]

X =

     4    10     0     0
     0     0    15     0
     1     0     0     4


X =

     4    10     0     0
     1     0    15     0
     1     0     0     4


Solution =

     4    10     0     0
     1     0    15     0
     1     0     0     4
```

```
OverallCost =

   114


ans =

     4    10     0     0
     1     0    15     0
     1     0     0     4
```

==

==

==

QUESTION – 2:(NORTH WEST CORNER METHOD)

	D1	D2	D3	SUPPLY
S1	2	7	4	5
S2	3	3	1	8
S3	5	4	7	7
S4	1	6	2	14
REQUIRED	7	9	18	34

```
clc;
clear all;
close all;
x=input('enter the transportation matrix');
disp(x);
```

```
     2     7     4     5
     3     3     1     8
     5     4     7     7
     1     6     2    14
     7     9    18    34
```

```
[m n]=size(x);
x1=zeros(m,n);
sumc=0;
sumr=0;
for i=1:m-1
 sumc=sumc+x(i,n);
end
for j=1:n-1
 sumr=sumr+x(m,j);
end
if(sumc == sumr)
 for i=1:m
 for j=1:n
 x11=min(x(i,n),x(m,j));
 x1(i,j)=x11;
 x(i,n)=x(i,n)-x11;
 x(m,j)=x(m,j)-x11;
 end
 end
else
    disp('unbalanced transportation');
end
 xre=0;
 for i=1:m-1
 for j=1:n-1
 xre=xre+(x(i,j).*x1(i,j));
 end
 end
 disp(['the transportation cost is ',num2str(xre)]);
```

```
the transportation cost is 102
```