# Basic code in C++ for UTA

#include <iostream>

#include <stdlib.h>

#include <math.h>

#include <time.h>

#include <ctime>

#include <fstream>

#include <cstdio>

#include <string>

using namespace std;

int main ()

{

ofstream fout("output-n.txt");

//............Time starts..............//

clock\_t time;

time = clock();

//...number of alternatives...........

int const N=5;

int M,t;

cout<<"Number of break points"<<endl;

cin>>M;

double S[M],K[N][M],W[N][M-1],s;

//....input of example......

int C[N]={-3,-4,-2,-6,-30};

//....new break points.......

t=10000000;

for (int i=0;i<N;i++)

if (C[i]<t)

{

t=C[i];

}

S[0]=t;

t=-10000000;

for (int i=0;i<N;i++)

if (C[i]>t)

{

t=C[i];

}

S[M-1]=t;

for (int i=1;i<M-1;i++)

S[i]=S[i-1]+float((S[M-1]-S[0])/(M-1));

for (int i=0;i<N;i++)

for (int j=0;j<M;j++)

K[i][j]=0;

for (int i=0;i<N;i++)

for (int j=0;j<M;j++)

{

if (C[i]>=S[j])

{

K[i][j]=1;

}

}

for (int i=0;i<N;i++)

for (int j=0;j<M-1;j++)

{

if ((K[i][j]-K[i][j+1])==1)

{

if (C[i]>S[j])

{

K[i][j]=K[i][j]+float((C[i]-S[j+1])/(S[j]-S[j+1]));

K[i][j+1]=float((C[i]-S[j])/(S[j+1]-S[j]));

}

}

}

//for (int j=0;j<M;j++)

// K[0][j]=0;

for (int i=0;i<N;i++)

K[i][0]=0;

//...Genretaing W..........

for (int i=0;i<N;i++)

for (int j=0;j<M-1;j++)

W[i][j]=K[i][j+1];

for (int i=0;i<N;i++)

for (int j=0;j<M-1;j++)

{

s=0;

for (int k=j;k<M-1;k++)

{

s=s+W[i][k];

W[i][j]=s;

}

}

for (int i=0;i<N;i++)

for (int j=0;j<M-1;j++)

if (W[i][j]>1)

W[i][j]=1;

fout<<"..........Break points.........."<<endl;

for (int i=0;i<M;i++)

fout<<S[i]<<endl;

fout<<"..........First equations........."<<endl;

for (int i=0;i<N;i++)

{

for (int j=0;j<M;j++)

fout<<K[i][j]<<"\t";

fout<<endl;

}

fout<<"..........Final equations........."<<endl;

for (int i=0;i<N;i++)

{

for (int j=0;j<M-1;j++)

cout<<W[i][j]<<"\t";

cout<<endl;

}

//............Calculation of Computational Time.........//

time = clock() - time;

cout << "It took :"<< ((float)time)/CLOCKS\_PER\_SEC << " seconds" << endl ;

return 0;

}

# Original UTA

#include <iostream>

#include <stdlib.h>

#include <math.h>

#include <time.h>

#include <ctime>

#include <fstream>

#include <cstdio>

#include <string>

using namespace std;

int main ()

{

Of stream fout("output-n.txt");

//............Time starts..............//

clock\_t time;

time = clock();

//...number of alternatives...........

int const N=5;

int M,t;

cout<<"Number of break points"<<endl;

cin>>M;

double S[M],K[N][M],W[N][M-1],s;

//....input of example......

int C[N]={-3,-4,-2,-6,-30};

//....new break points.......

t=10000000;

for (int i=0;i<N;i++)

if (C[i]<t)

{

t=C[i];

}

S[0]=t;

t=-10000000;

for (int i=0;i<N;i++)

if (C[i]>t)

{

t=C[i];

}

S[M-1]=t;

for (int i=1;i<M-1;i++)

S[i]=S[i-1]+float((S[M-1]-S[0])/(M-1));

for (int i=0;i<N;i++)

for (int j=0;j<M;j++)

K[i][j]=0;

for (int i=0;i<N;i++)

for (int j=0;j<M;j++)

{

if (C[i]>=S[j])

{

K[i][j]=1;

}

}

for (int i=0;i<N;i++)

for (int j=0;j<M-1;j++)

{

if ((K[i][j]-K[i][j+1])==1)

{

if (C[i]>S[j])

{

K[i][j]=K[i][j]+float((C[i]-S[j+1])/(S[j]-S[j+1]));

K[i][j+1]=float((C[i]-S[j])/(S[j+1]-S[j]));

}

}

}

//for (int j=0;j<M;j++)

// K[0][j]=0;

for (int i=0;i<N;i++)

K[i][0]=0;

//...Genretaing W..........

for (int i=0;i<N;i++)

for (int j=0;j<M-1;j++)

W[i][j]=K[i][j+1];

for (int i=0;i<N;i++)

for (int j=0;j<M-1;j++)

{

s=0;

for (int k=j;k<M-1;k++)

{

s=s+W[i][k];

W[i][j]=s;

}

}

for (int i=0;i<N;i++)

for (int j=0;j<M-1;j++)

if (W[i][j]>1)

W[i][j]=1;

fout<<"..........Break points.........."<<endl;

for (int i=0;i<M;i++)

fout<<S[i]<<endl;

fout<<"..........First equations........."<<endl;

for (int i=0;i<N;i++)

{

for (int j=0;j<M;j++)

fout<<K[i][j]<<"\t";

fout<<endl;

}

fout<<"..........Final equations........."<<endl;

for (int i=0;i<N;i++)

{

for (int j=0;j<M-1;j++)

cout<<W[i][j]<<"\t";

cout<<endl;

}

//............Calculation of Computational Time.........//

time = clock() - time;

cout << "It took :"<< ((float)time)/CLOCKS\_PER\_SEC << " seconds" << endl ;

return 0;

}

# Risk-averse UTA

MODEL:

DATA:

jj=7;

ENDDATA

SETS:

weight/1.. jj/;

snk(weight):w,s;

ENDSETS

! Objective function.....................................................................................................................................;

min=@sum(weight(j): w(j)\*s(j));

!Constraint 1 ...............................................................................................................................................;

@sum(weight(j):w(j))=1;

!Constraint 1 ...............................................................................................................................................;

w(1)>=W1L;

w(1)<=W1U;

!Constraint 1 ...............................................................................................................................................;

w(2)>= W2L;

w(2)<= W2U;

!Constraint 1 ...............................................................................................................................................;

w(3)>= W3L;

w(3)<= W3U;

!Constraint 1 ...............................................................................................................................................;

w(4)>= W4L;

w(4)<= W4U;

!Constraint 1 ...............................................................................................................................................;

w(5)>= W5L;

w(5)<= W5U;

!Constraint 1 ...............................................................................................................................................;

w(6)>= W6L;

w(6)<= W6U;

!Constraint 1 ...............................................................................................................................................;

w(7)>= W7L;

w(7)<= W7U;

!Data……………........................................................................................................................................;

DATA:

s =@ole('C:\FIVED.xlsx',FIVE1);

!Output........................................................................................................................................................;

@ole('C:\FIVED.xlsx',WEIGHTTT1)=W;

ENDDATA

**Optimization**

DATA:

ii=32;

jj=29;

ENDDATA

SETS:

Products/1.. ii/;

Areas/1.. jj/;

Alink(Products, Areas):x;

Blink(Products):D,w,p,a,s;

Dlink(Areas):WT,AT,PT;

ENDSETS

! Objective function.....................................................................................................................................;

Max=@sum(Products(i):@sum(Areas(j):s(i)\*x(i,j)));

!Constraint 1 ...............................................................................................................................................;

@for(Products(i):@sum(Areas(j):x(i,j))<=D(i));

!Constraint 2 ...............................................................................................................................................;

@for(Areas(j):@sum(Products(i):p(i)\*x(i,j))<=PT(j));

!Constraint 3 ...............................................................................................................................................;

@for(Areas(j):@sum(Products(i):w(i)\*x(i,j))<=WT(j));

!Constraint 4 ...............................................................................................................................................;

@for(Areas(j):@sum(Products(i):a(i)\*x(i,j))<=AT(j));

!Data……………........................................................................................................................................;

DATA:

D =@ole('C:\Data.xlsx',De\_i);

p =@ole('C:\Data.xlsx',h\_i);

w=@ole('C:\Data.xlsx',e\_i);

a =@ole('C:\Data.xlsx',w\_i);

s =@ole('C:\Data.xlsx',p\_i);

WT=@ole('C:\Data.xlsx',s\_j);

AT=@ole('C:\Data.xlsx',wt\_j);

PT=@ole('C:\Data.xlsx',ht\_j);

!Output........................................................................................................................................................;

@ole('E:\Data.xlsx',x\_ij)=x;

ENDDATA