www.optimyar.com

بر اساس پروتکل‌های دوره‌های آموزشی آپتیم‌یار، به اشتراک‌گذاری محتوا و کدهای نرم‌افزاری منظر حقوقی ممنوع است و از منظر اخلاقی نارضایتی مدرس دوره و گروه آپتیم‌یار را به همراه دارد.

از توجه شما به پروتکل دوره‌های آموزشی آپتیم‌یار سپاسگزاریم.

دوره جامع آنلاین بهینه‌سازی استوار و برنامه‌ریزی در شرایط عدم‌قطعیت همراه با کدنویسی در نرم‌افزار(GAMS)

Decision-Making under Uncertainty (Robust Optimization - Stochastic Programming - Fuzzy Programming)

مدرس:

دکتر علی پاپی (Ali Papi)

تخصص شاخص: بهینه‌سازی و تحقیق در عملیات، علم تحلیل داده، تکنیک‌های تجزیه و روش‌های حل دقیق، بهینه‌سازی استوار داده‌محور، هوش محاسباتی و الگوریتم‌های فرااِبتکاری، نظریه بازی، بهینه‌سازی چندهدفه و تصمیم‌گیری چندمعیاره

Optimization & Operations Research, Data Analytics, Computational Intelligence & Metaheuristics, Decomposition Techniques & Exact Methods, Data-Driven Robust Optimization, Game Theory, Multi Criteria Decision Making

کدهای نرم‌افزاری

SCND_Stochastic

BD_SCND_Stochastic

اخطار: بر اساس پروتکل‌های دوره‌های آموزشی آپتیم‌یار، به اشتراک‌گذاری محتوا و کدهای نرم‌افزاری منظر حقوقی ممنوع است و از منظر اخلاقی نارضایتی مدرس دوره و گروه آپتیم‌یار را به همراه دارد.

باز توجه شما به پروتکل دوره‌های آموزشی آپتیم‌یار سپاسگزاریم.

SCND_Stochastic

```
Sets
S       /s1*s10/
D       /d1*d20/
C       /c1*c30/
w       /w1*w5/
;


Parameters
A(s)
f(d)
b(s)
trSD(s,d)
trDC(d,c)
p
dem(c,w)
dem_N(c)
capD(d)
capS(s)
Prob(w)
/
w1      0.2
w2      0.3
w3      0.1
w4      0.2
w5      0.2
/
```

OR OptimYar

;

A(s) = uniform(1000,1500);

f(d) = uniform(2000,3000);

b(s) = uniform(5,10);

trSD(s,d)= uniform(1,2);

trDC(d,c)= uniform(0.5,0.7);

p = 15;

capD(d) = uniform(500,1000);

capS(s) = uniform(1000,2000);

dem(c,'w1') = uniform(50,100);

dem(c,'w2') = (1+0.8)*dem(c,'w1') ;

dem(c,'w3') = (1+0.2)*dem(c,'w1') ;

dem(c,'w4') = (1-0.2)*dem(c,'w1') ;

dem(c,'w5') = (1-0.3)*dem(c,'w1') ;

dem_N(c)= sum(w,prob(w)*dem(c,w));

Display

A

f

b

trSD

trDC

p

capD

capS

dem

dem_N

;

****

Free Variable

Z(w)

EB

;

Binary Variables

y(s)

x(d)

;

Positive Variable

u(s,w)

QSD(s,d,w)

QDC(d,c,w)

;

Equations

```
obj_RNSSP
cons1
cons2
cons3
cons4
cons5
cons6
cons7

;

obj_RNSSP..      EB =e= sum(w,Prob(w)*
 [ p*sum({d,c},QDC(d,c,w)) -  sum({s,d},trSD(s,d)*QSD(s,d,w))
- sum({d,c},trDC(d,c)*QDC(d,c,w)) - sum(s,b(s)*u(s,w))   ]
 )

-
(sum(d,f(d)*x(d)) + sum(s,A(s)*y(s)))
;


cons1(s,w)..       u(s,w) =L= capS(s)*y(s);

cons2(d,w)..       sum(S,QSD(s,d,w))=L= capD(d)*x(d);

cons3(s,w)..       u(s,w) - sum(d,QSD(s,d,w))=g= 0  ;

cons4(d,w)..       sum(s,QSD(s,d,w)) - sum(c,QDC(d,c,w)) =g= 0;
```

```
cons5(c,w)..      sum(d,QDC(d,c,w)) =l= dem(c,w);

cons6(c,w)..      sum(d,QDC(d,c,w)) =g= 3/10*dem(c,w);

cons7..           sum(d,x(d)) =l= card(d)/2;



Model SCND
/
obj_RNSSP
cons1
cons2
cons3
cons4
cons5
cons6
cons7
/
;

Options
mip = CPLEX
reslim =100
optcr = 0
;
```

```
Solve SCND us mip max EB;


Display
EB.l
y.l
x.l
QSD.l
QDC.l
;
```

BD_SCND_Stochastic

********** BD for SCND **********

****

********** Data **********

```
Sets
S       /s1*s10/
D       /d1*d20/
C       /c1*c30/
w       /w1*w5/
;

Parameters
A(s)
f(d)
b(s)
trSD(s,d)
trDC(d,c)
p
dem(c,w)
dem_N(c)
capD(d)
capS(s)
Prob(w)
/
w1      0.2
w2      0.3
```

```
w3      0.1
w4      0.2
w5      0.2
/


;



A(s)    = uniform(1000,1500);
f(d)    = uniform(2000,3000);
b(s)    = uniform(5,10);
trSD(s,d)= uniform(1,2);
trDC(d,c)= uniform(0.5,0.7);
p       = 15;
capD(d)  = uniform(500,1000);
capS(s)  = uniform(1000,2000);



dem(c,'w1')   = uniform(50,100);
dem(c,'w2')   = (1+0.8)*dem(c,'w1') ;
dem(c,'w3')   = (1+0.2)*dem(c,'w1') ;
dem(c,'w4')   = (1-0.2)*dem(c,'w1') ;
dem(c,'w5')   = (1-0.3)*dem(c,'w1') ;


dem_N(c)= sum(w,prob(w)*dem(c,w));



Display
```

A

f

b

trSD

trDC

p

capD

capS

dem

dem_N

;

********** End of Data **********

********** DSP **********

Free Variable

Z_DSP

;

Parameters

yL(s)

xL(d)

;

Positive Variables

L1_w(s)

L2_w(d)

L5_w(c)

;

Negative Variables

L3_w(s)

L4_w(d)

L6_w(c)

;

Parameter

dem_w(c)

Prob_w

;

Equation

objDSP

Eq_u_DSP

Eq_QSD_DSP

Eq_QDC_DSP

;

```
objDSP.. Z_DSP =e=


sum(s,capS(s)*yL(s)*L1_w(s))
+
sum(d,capD(d)*xL(d)*L2_w(d))
+
sum(c,dem_w(c)*L5_w(c))
+
sum(c,3/10*dem_w(c)*L6_w(c))
;


Eq_u_DSP(s)..          L1_w(s) + L3_w(s) =g=  - Prob_w*b(s) ;


Eq_QSD_DSP(s,d)..      L2_w(d) - L3_w(s) + L4_w(d) =g= -Prob_w*trSD(s,d);


Eq_QDC_DSP(d,c)..      - L4_w(d) + L5_w(c) + L6_w(c) =g=  Prob_w*p -Prob_w*trDC(d,c);




Model DSP
/
objDSP
Eq_u_DSP
Eq_QSD_DSP
Eq_QDC_DSP
/
;
```

********** End of DSP **********

********** MSP **********

Free Variable

Z_MSP

;

Equation

objMSP

Eq_u_MSP

Eq_QSD_MSP

Eq_QDC_MSP

Eq_Normalized_MSP_1

Eq_Normalized_MSP_2

;

objMSP.. Z_MSP =e=

sum(s,capS(s)*yL(s)*L1_w(s))

```
+
sum(d,capD(d)*xL(d)*L2_w(d))
+
sum(c,dem_w(c)*L5_w(c))
+
sum(c,3/10*dem_w(c)*L6_w(c))
;


Eq_u_MSP(s)..          L1_w(s) + L3_w(s) =g=  0 ;


Eq_QSD_MSP(s,d)..      L2_w(d) - L3_w(s) + L4_w(d) =g= 0;


Eq_QDC_MSP(d,c)..     - L4_w(d) + L5_w(c) + L6_w(c) =g=  0;


Eq_Normalized_MSP_1..    sum(s,L1_w(s) ) + sum(d,L2_w(d) ) +  sum(c, L5_w(c))  =e= 1 ;
Eq_Normalized_MSP_2..    sum(s,L3_w(s) ) + sum(d,L4_w(d) ) +  sum(c,L6_w(c))   =e= -1 ;
;


Model MSP
/
objMSP
Eq_u_MSP
Eq_QSD_MSP
Eq_QDC_MSP
Eq_Normalized_MSP_1
Eq_Normalized_MSP_2
/
;
```

****

********** End of MSP **********

****

********** RMP **********

****

Binary Variables

y(s)

x(d)

;

Free Variable

Z_RMP

say

;

say.up=1e7;

Equations

obj_RMP

cons7_RMP

```
OptimalityCut
FeasibilityCut
;

Set iter /iter1*iter50/;

Set
OC(iter)
FC(iter,w)
;
OC(Iter) = NO ;
FC(iter,w)= NO ;

Parameters
L1(s,iter,w)
L2(d,iter,w)
L5(c,iter,w)
L3(s,iter,w)
L4(d,iter,w)
L6(c,iter,w)
;

Parameters
demm(c,iter,w)
;
demm(c,iter,w)=dem(c,w) ;
```

OR OptimYar

```
Equations
obj_RMP
cons7_RMP
OptimalityCut
FeasibilityCut
;




obj_RMP ..          Z_RMP  =e= -  ( sum(d,f(d)*x(d)) + sum(s,A(s)*y(s)) ) +  say ;

cons7_RMP..         sum(d,x(d)) =l= card(d)/2;

OptimalityCut(OC)..   say =l=
sum(w,


sum(s,capS(s)*y(s)*L1(s,oc,w))
+
sum(d,capD(d)*x(d)*L2(d,oc,w))
+
sum(c,dem(c,w)*L5(c,oc,w))
+
sum(c,3/10*dem(c,w)*L6(c,oc,w))
)


;
FeasibilityCut(FC)..
```

```
sum(s,capS(s)*y(s)*L1(s,fc))
+
sum(d,capD(d)*x(d)*L2(d,fc))
+
sum(c,demm(c,fc)*L5(c,fc))
+
sum(c,3/10*demm(c,fc)*L6(c,fc))


=g= 0;



Model RMP
/
obj_RMP
cons7_RMP
OptimalityCut
FeasibilityCut
/
;


************************************************************************************
****************** End of RMP ******************************************************


************************************************************************************
******************** Main Loop of BD ***********************************************
```

****

Options

MIP = CPLEX

LP = CPLEX

RESLIM = 100

OPTCR = 0

;

Initial Value*******

yL('s1')=1;

xL('d1')=1;

** Max of Relative Error

Scalar Max_RE;

Max_RE = 0.001;

Scalar RE;

Scalar Convergency ;

Convergency =NO;

****

****

Scalar

UB

LB

;

UB = inf ;

```
LB = -inf ;
;



Parameter Result(iter,*);



Scalar Bounded;



Loop(iter$(NOT(Convergency)),

Bounded = YES;
Loop(w,

Prob_w=Prob(w);
dem_w(c)=dem(c,w);

Solve DSP us LP min Z_DSP;
abort$(DSP.ModelStat=2) "Your MP Model is not fasible"  ;



if(DSP.ModelStat <> 3,
```

```
L1(s,iter,w) = L1_w.l(s) ;
L2(d,iter,w) = L2_w.l(d) ;
L5(c,iter,w) = L5_w.l(c) ;
L3(s,iter,w) = L3_w.l(s) ;
L4(d,iter,w) = L4_w.l(d) ;
L6(c,iter,w) = L6_w.l(c) ;



else

Bounded = NO;

Solve MSP us LP min Z_MSP;

L1(s,iter,w) = L1_w.l(s) ;
L2(d,iter,w) = L2_w.l(d) ;
L5(c,iter,w) = L5_w.l(c) ;
L3(s,iter,w) = L3_w.l(s) ;
L4(d,iter,w) = L4_w.l(d) ;
L6(c,iter,w) = L6_w.l(c) ;

FC(Iter,w) = YES;
Result(iter,'FC')= YES;


)
;
```

```
);
*End of Loop for the Scenarios

if(Bounded = 1,
OC(Iter) = YES ;

LB=

sum(w,

sum(s,capS(s)*yL(s)*L1(s,iter,w))
+
sum(d,capD(d)*xL(d)*L2(d,iter,w))
+
sum(c,dem(c,w)*L5(c,iter,w))
+
sum(c,3/10*dem(c,w)*L6(c,iter,w))
)

- ( sum(d,f(d)*xL(d)) + sum(s,A(s)*yL(s)) )

;


Result(iter,'LB')= LB;
Result(iter,'OC')= YES;
```

OR OptimYar

```
)
;



**********
Solve RMP us MIP max Z_RMP;
abort$(RMP.ModelStat=2) "Your MP Model is not fasible"  ;


yL(s) = y.l(s);
xL(d) = x.l(d);


UB=Z_RMP.l;


Result(iter,'UB')= Z_RMP.l;
**********



******* Stop Criterion **********


RE = abs((UB-LB)/UB) ;


if( RE <= Max_RE,
Convergency = YES;
)
;


Result(iter,'RE')= RE;
```

****

);

*End of Loop

Display

Result

y.l

x.l

;

www.optimyar.com

دوره جامع آنلاین بهینه‌سازی استوار و برنامه‌ریزی در شرایط عدم‌قطعیت همراه با کدنویسی در نرم‌افزار(GAMS)

Decision-Making under Uncertainty (Robust Optimization - Stochastic Programming - Fuzzy Programming)

مدرس:

دکتر علی پاپی (Ali Papi)

تخصص شاخص: بهینه‌سازی و تحقیق در عملیات، علم تحلیل داده، تکنیک‌های تجزیه و روش‌های حل دقیق، بهینه‌سازی استوار داده‌محور، هوش محاسباتی و الگوریتم‌های فرابتکاری، نظریه بازی، بهینه‌سازی چندهدفه و تصمیم‌گیری چندمعیاره

Optimization & Operations Research, Data Analytics, Computational Intelligence & Metaheuristics, Decomposition Techniques & Exact Methods, Data-Driven Robust Optimization, Game Theory, Multi Criteria Decision Making