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 فراابتکاری، 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

www.optimyar.com

کدهای نرم‌افزاری

RobustMODM

Robust_Payoff

Robust_TH

Robust_TH_SensitivityAnalysis

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.

پاز توجه شما به پروتکل دوره‌های آموزشی آپتیم‌یار سپاسگزاریم.

RobustMODM

Sets

/j x /j1*j100

/k y /k1*k20

/i cons /i1*i50

Parameters

'c(j)'nominal

f(k)

r(j)

'a(i,j) 'nominal

d(i,k)

b(i)

h(k)

'e 'nominal

;

;(10,20)c(j)   = uniform

;(0,50)r(j)   = uniform

;(700,1000)f(k)   = uniform

;(2,8)a(i,j)   = uniform

;(300,500)d(i,k)   = uniform

;(1500,2500)b(i)   = uniform

;(1,3)h(k)   = uniform

```
;e       = 10
;

Parameters
PR_C(j)
PR_a(i,j)
PR_e
;

;PR_C(j) = 0.30
;PR_a(i,j) = 0.50
;PR_e = 0.60

Scalars
Gamma_o
Gamma_c1
Gamma_c2
;

;Gamma_o  = sqrt(card(j))
;Gamma_c1 = sqrt(card(i)*card(j))
;Gamma_c2 = 1


Positive Variables
x(j)
;
```

```
Binary Variables
y(k)
;

Free Variables
"Z1  "min
"Z2  "max
;



Equations
obj1
obj2
obj_RC
cons1
cons1_RC
cons2
cons_add
;

Positive variables
p_o(j)
q_o

p_c1(i,j)
;q_c1(i)
;
```

```
min❋

;   obj1..    Z1 =g=  sum(j,c(j)*x(j)) + sum(k,f(k)*y(k)) +  sum(j,p_o(j) ) + Gamma_o*q_o


;obj_RC(j)..  p_o(j) + q_o =g= x(j)*c(j)*PR_c(j)


max❋

;obj2..    Z2 =e=  sum(j,r(j)*x(j))


; cons1(i)..     sum(j,a(i,j)*x(j)) + sum(k,d(i,k)*y(k)) - (sum(j,p_c1(i,j) ) + Gamma_c1*q_c1(i) ) =g= b(i)


; cons_add(i)..  sum(j,a(i,j)*(1-PR_a(i,j))*x(j)) =l= b(i)


;cons1_RC(i,j).. p_c1(i,j) + q_c1(i) =g= x(j)*a(i,j)* PR_a(i,j)


; cons2..        sum(k,h(k)*y(k)) =l= e - Gamma_c2*e*PR_e


Model BS_RC
/
obj1
obj2
obj_RC
cons1
cons1_RC
cons2
cons_add
/
```

```
;

Options
MIP = CPLEX
OPTCR =0
RESLIM = 100
;

; Solve   BS_RC us MIP min Z1
Display
"Focus on min Z1"
z1.l
z2.l
x.l
y.l
;
; Solve   BS_RC us MIP max Z2
Display
"Focus on max Z2"
z1.l
z2.l
x.l
y.l
;
```

Robust_Payoff

```
******************************
******* The Best Payoff Matrix
******************************
Set Objs
/
of1
of2
/
;
Alias(Objs,of)
;

Parameter
Payoff(of,of)
Max_o(of)
Min_o(of)
R_o(of)
;

****************** Problem Formulation/Modeling *******************************

Sets

j x /j1*j100/
k y /k1*k20/
i cons  /i1*i50/
```

```
Parameters
c(j)'nominal'
f(k)
r(j)
a(i,j)  'nominal'
d(i,k)
b(i)
h(k)
e  'nominal'
;

c(j)     = uniform(10,20);
r(j)     = uniform(0,50);
f(k)     = uniform(700,1000);
a(i,j)   = uniform(2,8);
d(i,k)   = uniform(300,500);
b(i)     = uniform(1500,2500);
h(k)     = uniform(1,3);
e        = 10;
;

Parameters
PR_C(j)
PR_a(i,j)
PR_e
;

PR_C(j) = 0.30;
PR_a(i,j) = 0.50;
```

```
PR_e = 0.60;


Scalars
Gamma_o
Gamma_c1
Gamma_c2
;


Gamma_o  = sqrt(card(j));
Gamma_c1 = sqrt(card(i)*card(j));
Gamma_c2 = 1;


Positive Variables
x(j)
;


Binary Variables
y(k)
;


Free Variables
Z1   "min"
Z2   "max"
;
```

```
Equations
obj1
obj2
obj_RC
cons1
cons1_RC
cons2
cons_add
;

Positive variables
p_o(j)
q_o

p_c1(i,j)
q_c1(i);
;

*min
obj1..    Z1 =g=  sum(j,c(j)*x(j)) + sum(k,f(k)*y(k)) +  sum(j,p_o(j) ) + Gamma_o*q_o   ;

obj_RC(j)..  p_o(j) + q_o =g= x(j)*c(j)*PR_c(j);

*max
obj2..    Z2 =e=  sum(j,r(j)*x(j));
```

```
cons1(i)..     sum(j,a(i,j)*x(j)) + sum(k,d(i,k)*y(k)) - (sum(j,p_c1(i,j) ) + Gamma_c1*q_c1(i) ) =g= b(i) ;

cons_add(i)..  sum(j,a(i,j)*(1-PR_a(i,j))*x(j)) =l= b(i) ;

cons1_RC(i,j).. p_c1(i,j) + q_c1(i) =g= x(j)*a(i,j)* PR_a(i,j);

cons2..        sum(k,h(k)*y(k)) =l= e - Gamma_c2*e*PR_e ;

Model BS_RC
/
obj1
obj2
obj_RC
cons1
cons1_RC
cons2
cons_add
/
;

Options
MIP = CPLEX
OPTCR =0
RESLIM = 100
;
```

```
******************************** Find the "Payoff" matrix***********************


Solve   BS_RC us MIP min Z1 ;
Payoff('of1','of1') = Z1.l;


Solve   BS_RC us MIP max Z2 ;
Payoff('of2','of2') = Z2.l;


Z2.fx=Payoff('of2','of2');
Solve   BS_RC us MIP min Z1 ;
Payoff('of1','of2') = Z1.l;
Z2.lo=-inf ;
Z2.up=inf ;



Z1.fx=Payoff('of1','of1');
Solve   BS_RC us MIP max Z2 ;
Payoff('of2','of1') = Z2.l;
Z1.lo=-inf ;
Z1.up=inf  ;


******************************************  Min Max Range


Min_o(of)= smin[objs,payoff(of,objs)];
Max_o(of)= smax[objs,payoff(of,objs)];
R_o(of)= Max_o(of) - Min_o(of) ;


**********************************************************
```

Display

Payoff

Min_o

Max_o

R_o

Robust_TH

```
* MODM Method (TH) A. Papi
* Trading-off between Compensatory (Norm 1) and Non-compensatory Solutions (Norm  inf)



***************** Problem Formulation/Modeling ******************************


Sets


j x /j1*j100/
k y /k1*k20/
i cons  /i1*i50/


Parameters
c(j)'nominal'
f(k)
r(j)
a(i,j)  'nominal'
d(i,k)
b(i)
h(k)
e  'nominal'
;


c(j)     = uniform(10,20);
r(j)     = uniform(0,50);
f(k)     = uniform(700,1000);
```

```
a(i,j)   = uniform(2,8);
d(i,k)   = uniform(300,500);
b(i)     = uniform(1500,2500);
h(k)     = uniform(1,3);
e        = 10;
;

Parameters
PR_C(j)
PR_a(i,j)
PR_e
;

PR_C(j) = 0.30;
PR_a(i,j) = 0.50;
PR_e = 0.60;

Scalars
Gamma_o
Gamma_c1
Gamma_c2
;

Gamma_o  = sqrt(card(j));
Gamma_c1 = sqrt(card(i)*card(j));
Gamma_c2 = 1;
```

```
Positive Variables
x(j)
;

Binary Variables
y(k)
;

Free Variables
Z1   "min"
Z2   "max"
;

Equations
obj1
obj2
obj_RC
cons1
cons1_RC
cons2
cons_add
;

Positive variables
p_o(j)
q_o
```

```
p_c1(i,j)
q_c1(i);
;



*min
obj1..    Z1 =g=  sum(j,c(j)*x(j)) + sum(k,f(k)*y(k)) +  sum(j,p_o(j) ) + Gamma_o*q_o   ;


obj_RC(j)..  p_o(j) + q_o =g= x(j)*c(j)*PR_c(j);


*max
obj2..    Z2 =e=  sum(j,r(j)*x(j));


cons1(i)..     sum(j,a(i,j)*x(j)) + sum(k,d(i,k)*y(k)) - (sum(j,p_c1(i,j) ) + Gamma_c1*q_c1(i) ) =g= b(i) ;


cons_add(i)..  sum(j,a(i,j)*(1-PR_a(i,j))*x(j)) =l= b(i) ;



cons1_RC(i,j).. p_c1(i,j) + q_c1(i) =g= x(j)*a(i,j)* PR_a(i,j);



cons2..       sum(k,h(k)*y(k)) =l= e - Gamma_c2*e*PR_e ;


Model BS_RC
/
obj1
```

```
obj2
obj_RC
cons1
cons1_RC
cons2
cons_add
/
;

Options
MIP = CPLEX
OPTCR =0
RESLIM = 100
;

******************************
******* The Best Payoff Matrix
******************************
Set Objs
/
of1
of2
/
;
Alias(Objs,of)
;

Parameter
Payoff(of,of)
```

```
Max_o(of)

Min_o(of)

R_o(of)

;


******************************** Find the "Payoff" matrix***********************


Solve   BS_RC us MIP min Z1 ;

Payoff('of1','of1') = Z1.l;


Solve   BS_RC us MIP max Z2 ;

Payoff('of2','of2') = Z2.l;


Z2.fx=Payoff('of2','of2');

Solve   BS_RC us MIP min Z1 ;

Payoff('of1','of2') = Z1.l;

Z2.lo=-inf ;

Z2.up=inf ;



Z1.fx=Payoff('of1','of1');

Solve   BS_RC us MIP max Z2 ;

Payoff('of2','of1') = Z2.l;

Z1.lo=-inf ;

Z1.up=inf  ;


******************************************   Min Max Range
```

```
Min_o(of)= smin[objs,payoff(of,objs)];

Max_o(of)= smax[objs,payoff(of,objs)];

R_o(of)= Max_o(of) - Min_o(of) ;


*************************************************************


Display

Payoff

Min_o

Max_o

R_o

*************************************************************


** TH Method


Positive Variable

N1

N2

;


Equations

Norm_of1  'min : the least is the best'

Norm_of2  'max : the least is the best'

;


Norm_of1..      N1 =e=  [Z1 - min_o('of1')]/R_o('of1');

Norm_of2..      N2 =e=  [max_o('of2') - Z2]/R_o('of2');
```

```
Positive Variable
Lp1
Lpinf


;
Equations
E_Lp1
E_Lpinf_of1
E_Lpinf_of2
;


E_Lp1 ..          Lp1 =e=  [N1+N2]/2;
E_Lpinf_of1..     Lpinf =g=  N1;
E_Lpinf_of2..     Lpinf =g=  N2;



********** TH Measure
Equations
Me_TH
;
Free Variable Z_TH  'min'
;
Scalars
w1 /0.999/
w2 /0.001/
;
                        1
Me_TH   ..        Z_TH =e=  w1*Lp1 + w2*Lpinf;
```

```
Model Robust_TH
/
BS_RC
Norm_of1
Norm_of2
E_Lp1
E_Lpinf_of1
E_Lpinf_of2
Me_TH
/
;

Solve Robust_TH us MIP min  Z_TH;

********** Final Result/Output******************************************************
Display
z1.l
z2.l
N1.l
N2.l
Lp1.l
Lpinf.l
x.l
y.l
;
```

Robust_TH_SensitivityAnalysis

```
* MODM Method (TH) A. Papi
* Trading-off between Compensatory (Norm 1) and Non-compensatory Solutions (Norm  inf)



****************** Problem Formulation/Modeling ********************************


Sets


j x /j1*j100/
k y /k1*k20/
i cons  /i1*i50/


Parameters
c(j)'nominal'
f(k)
r(j)
a(i,j)  'nominal'
d(i,k)
b(i)
h(k)
e  'nominal'
;


c(j)     = uniform(10,20);
r(j)     = uniform(0,50);
f(k)     = uniform(700,1000);
```

```
a(i,j)   = uniform(2,8);
d(i,k)   = uniform(300,500);
b(i)     = uniform(1500,2500);
h(k)     = uniform(1,3);
e        = 10;
;

Parameters
PR_C(j)
PR_a(i,j)
PR_e
;

PR_C(j) = 0.30;
PR_a(i,j) = 0.50;
PR_e = 0.60;

Scalars
Gamma_o
Gamma_c1
Gamma_c2
;

Gamma_o  = sqrt(card(j));
Gamma_c1 = sqrt(card(i)*card(j));
Gamma_c2 = 1;
```

```
Positive Variables
x(j)
;

Binary Variables
y(k)
;

Free Variables
Z1   "min"
Z2   "max"
;

Equations
obj1
obj2
obj_RC
cons1
cons1_RC
cons2
cons_add
;

Positive variables
p_o(j)
q_o
```

```
p_c1(i,j)
q_c1(i);
;

*min
obj1..    Z1 =g=  sum(j,c(j)*x(j)) + sum(k,f(k)*y(k)) +  sum(j,p_o(j) ) + Gamma_o*q_o   ;

obj_RC(j)..  p_o(j) + q_o =g= x(j)*c(j)*PR_c(j);

*max
obj2..    Z2 =e=  sum(j,r(j)*x(j));

cons1(i)..     sum(j,a(i,j)*x(j)) + sum(k,d(i,k)*y(k)) - (sum(j,p_c1(i,j) ) + Gamma_c1*q_c1(i) ) =g= b(i) ;

cons_add(i)..  sum(j,a(i,j)*(1-PR_a(i,j))*x(j)) =l= b(i) ;

cons1_RC(i,j).. p_c1(i,j) + q_c1(i) =g= x(j)*a(i,j)* PR_a(i,j);

cons2..        sum(k,h(k)*y(k)) =l= e - Gamma_c2*e*PR_e ;

Model BS_RC
/
obj1
```

```
obj2
obj_RC
cons1
cons1_RC
cons2
cons_add
/
;

Options
MIP = CPLEX
OPTCR =0
RESLIM = 100
;

******************************
******* The Best Payoff Matrix
******************************
Set Objs
/
of1
of2
/
;
Alias(Objs,of)
;

Set Iter /iter1*iter5/
```

```
Parameter
Payoff(of,of,iter)
Max_o(of,iter)
Min_o(of,iter)
R_o(of,iter)
;

******************************** Find the "Payoff" matrix**********************

Parameter
teta(iter)
/
iter1 -0.2
iter2 -0.1
iter3  0.1
iter4  0.2
iter5  0.3
/
;

Parameter
f_SA(k,iter);
f_SA(k,iter)=(1+teta(iter))*f(k);

Loop(iter,

f(k)=f_SA(k,iter);
```

```
Solve   BS_RC us MIP min Z1 ;
Payoff('of1','of1',iter) = Z1.l;

Solve   BS_RC us MIP max Z2 ;
Payoff('of2','of2',iter) = Z2.l;

Z2.fx=Payoff('of2','of2',iter);
Solve   BS_RC us MIP min Z1 ;
Payoff('of1','of2',iter) = Z1.l;
Z2.lo=-inf ;
Z2.up=inf ;

Z1.fx=Payoff('of1','of1',iter);
Solve   BS_RC us MIP max Z2 ;
Payoff('of2','of1',iter) = Z2.l;
Z1.lo=-inf ;
Z1.up=inf  ;

)
;

******************************************   Min Max Range

Min_o(of,iter)= smin[objs,payoff(of,objs,iter)];
Max_o(of,iter)= smax[objs,payoff(of,objs,iter)];
R_o(of,iter)= Max_o(of,iter) - Min_o(of,iter) ;
```

```
**********************************************************

Display
Payoff
Min_o
Max_o
R_o
**********************************************************


** TH Method


Parameters
min_o_iter(of)
max_o_iter(of)
R_o_iter(of)
;


Positive Variable
N1
N2
;


Equations
```

```
Norm_of1  'min : the least is the best'
Norm_of2  'max : the least is the best'
;


Norm_of1..      N1 =e=  [Z1 - min_o_iter('of1')]/R_o_iter('of1');
Norm_of2..      N2 =e=  [max_o_iter('of2') - Z2]/R_o_iter('of2');


Positive Variable
Lp1
Lpinf


;
Equations
E_Lp1
E_Lpinf_of1
E_Lpinf_of2
;


E_Lp1 ..          Lp1 =e=  [N1+N2]/2;
E_Lpinf_of1..     Lpinf =g=  N1;
E_Lpinf_of2..     Lpinf =g=  N2;



********** TH Measure
Equations
Me_TH
;
Free Variable Z_TH  'min'
```

```
;
Scalars
w1 /0.999/
w2 /0.001/
;

Me_TH   ..       Z_TH =e=  w1*Lp1 + w2*Lpinf;



Model Robust_TH
/
BS_RC
Norm_of1
Norm_of2
E_Lp1
E_Lpinf_of1
E_Lpinf_of2
Me_TH
/
;


Loop(iter,
min_o_iter(of)=min_o(of,iter);
max_o_iter(of)=max_o(of,iter);
R_o_iter(of)=R_o(of,iter);

Solve Robust_TH us MIP min  Z_TH;
```

********** Final Result/Output**

Display

'iter'

iter

z1.l

z2.l

N1.l

N2.l

Lp1.l

Lpinf.l

x.l

y.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 فراابتکاری، 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