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**

**Main_StochasticProgrammin_Benders**

**DATA_StochasticProgramming_BD**

**DSP_StochasticProgramming_BD**

**MSP_StochasticProgramming_BD**

**RMP_StochasticProgramming_BD**

**BD_StochasticProgramming_Implementation**

**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.

# Main_StochasticProgrammin_Benders

```
Sets


j x /j1*j100/
i y /i1*i20/
n cons1  /n1*n50/
m cons2  /m1*m20/
Scen     /scen1*scen5/
;


Parameter
Prob(Scen);


Prob(Scen)=1/card(Scen);




Parameters
c(j)
f(i)
a(n,j)
d(n,i)
b_N(n)
b(n,Scen)
k(m,i)
e(m)
```

;

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

f(i) = uniform(-1000,-700);

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

d(n,i) = uniform(-500,-300);

b_N(n) = uniform(-1500,0);

k(m,i) = uniform(2,3);

e(m) = uniform(20,30);

;

b(n,'Scen1') = (1-9/10)*b_N(n);

b(n,'Scen2') = (1-3/10)*b_N(n);

b(n,'Scen3') = (1)*b_N(n);

b(n,'Scen4') = (1+3/10)*b_N(n);

b(n,'Scen5') = (1+9/10)*b_N(n);

Positive Variables

x(j,Scen)

;

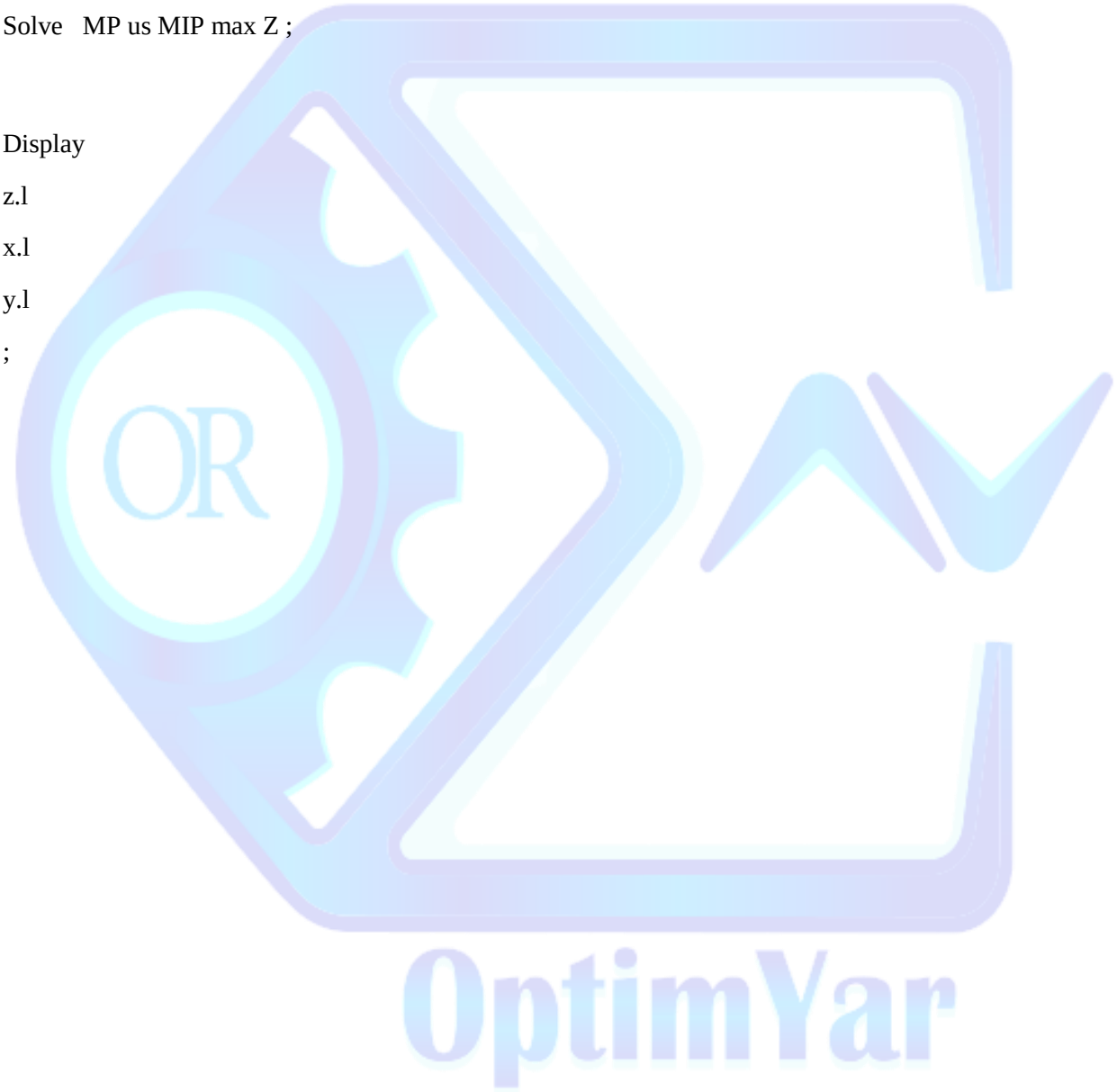
Binary Variables

y(i)

;

```
Free Variables
Z
;


Equations
obj
cons1
cons2
;

obj..        z =e=  sum(Scen, Prob(Scen)*[sum(j,c(j)*x(j,Scen))] ) + sum(i,f(i)*y(i)) ;

cons1(n,Scen)..     sum(j,a(n,j)*x(j,Scen)) + sum(i,d(n,i)*y(i)) =l= b(n,Scen) ;

cons2(m)..     sum(i,k(m,i)*y(i)) =l= e(m) ;

Model MP
/
obj
cons1
cons2
/
;

Options
MIP = CPLEX
OPTCR =0
```

```
RESLIM = 100
;



Solve   MP us MIP max Z ;


Display
z.l
x.l
y.l
;
```



# DATA_StochasticProgramming_BD

Sets

j x /j1*j100/

i y /i1*i20/

n cons1  /n1*n50/

m cons2  /m1*m20/

Scen  /scen1*scen5/

;

Parameter

Prob(Scen);

Prob(Scen)=1/card(Scen);

Parameters

c(j)

f(i)

a(n,j)

d(n,i)

b_N(n)

b(n,Scen)

k(m,i)

```
e(m)
;


c(j)     = uniform(10,20);
f(i)     = uniform(-1000,-700);
a(n,j)   = uniform(2,8);
d(n,i)   = uniform(-500,-300);
b_N(n)     = uniform(-1500,0);
k(m,i)   = uniform(2,3);
e(m)     = uniform(20,30);
;


b(n,'Scen1') = (1-9/10)*b_N(n);
b(n,'Scen2') = (1-3/10)*b_N(n);
b(n,'Scen3') = (1)*b_N(n);
b(n,'Scen4') = (1+3/10)*b_N(n);
b(n,'Scen5') = (1+9/10)*b_N(n);




Display
c
f
a
d
b_N
b
k
```

OptimYar

e

prob

**********

# DSP_StochasticProgramming_BD

```
*$include DATA_BD.gms
**************************************************************************
********** Dual Sub Problem (DSP) of Benders Decompostion *********
**************************************************************************


Positive Variables
u(n)
;


Parameters
yL(i)
;



Free Variable
Z_DSP
;



Equations
obj_DSP
cons1_DSP
;


Parameters
```

```
Prob_s
b_s(n)
;


obj_DSP..          Z_DSP =e= sum(n,[b_s(n) - sum(i,d(n,i)*yL(i))]*u(n))  ;
cons1_DSP(j)..     sum(n,a(n,j)*u(n))  =g= Prob_s*c(j) ;


Model DSP
/
obj_DSP
cons1_DSP
/
;

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

# MSP_StochasticProgramming_BD

```
*$include DATA_BD.gms
**************************************************************************
********** Modified Sub Problem (MSP) of Benders Decompostion *****
**************************************************************************


Positive Variables
u(n)
;


Parameters
yL(i)
;



Free Variable
Z_MSP
;



Equations
obj_MSP
cons1_MSP
cons2_MSP
;
```

```
obj_MSP..          Z_MSP =e= sum(n,[b_s(n) - sum(i,d(n,i)*yL(i))]*u(n));


cons1_MSP(j)..     sum(n,a(n,j)*u(n))  =g= 0 ;


cons2_MSP..        sum(n,u(n))  =e= 1 ;




Model MSP
/
obj_MSP
cons1_MSP
cons2_MSP
/
;


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

OR OptimYar

# RMP_StochasticProgramming_BD

```
*$include DATA_BD.gms
**************************************************************************
********** Ralaxed Master Problem (RMP) of Benders Decompostion ***
**************************************************************************


Binary Variable
y(i)
;


Free Variable
Z_RMP
say
;

say.up=1e6;


Equations
obj_RMP
cons2_RMP
OptimalityCut
FeasibilityCut
;
```

```
Set
OC(iter)
FC(iter,scen)
;
OC(Iter) = NO ;
FC(iter,scen)= NO ;



Parameter
uL(n,Iter,scen)
;



Parameters  bb(n,iter,scen);
bb(n,iter,scen)=b(n,scen) ;



obj_RMP ..          z_RMP  =e=sum(i,f(i)*y(i)) + say ;
cons2_RMP(m)..  sum(i,k(m,i)*y(i)) =l= e(m) ;

OptimalityCut(OC)..   say =l=   sum(scen, [sum(n,[b(n,scen) - sum(i,d(n,i)*y(i))]*uL(n,oc,scen) )]);
FeasibilityCut(FC)..  sum(n,[bb(n,fc) - sum(i,d(n,i)*y(i))]*uL(n,fc) ) =g= 0;



Model RMP
/
```

```
obj_RMP
cons2_RMP
OptimalityCut
FeasibilityCut
/
;


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

# BD_StochasticProgramming_Implementation

********** Implementaion of Benders Decomposition**********

**********

```
$include DATA_StochasticProgramming_BD.gms
$include DSP_StochasticProgramming_BD.gms
$include MSP_StochasticProgramming_BD.gms
Set iter /iter1*iter20/;
$include RMP_StochasticProgramming_BD.gms
```

**********

**Initial Value********

```
yL(i)=0;
```

** Max of Relative Error

```
Scalar Max_RE;
Max_RE = 0.001;
Scalar RE;
```

```
Scalar Convergency ;
Convergency =NO;
```

**********

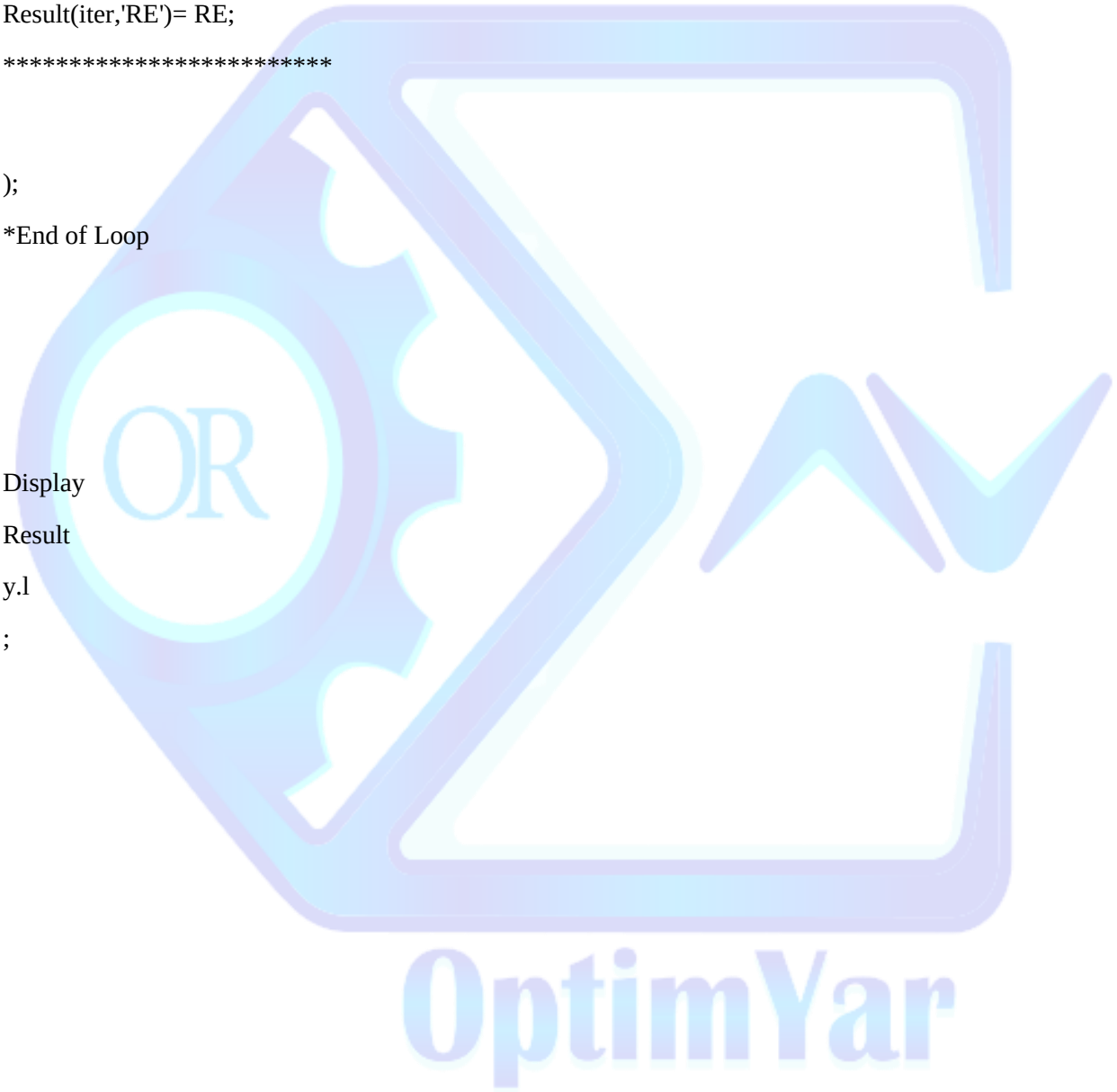
**********

```
Scalar
UB
LB
;
UB = inf ;
LB = -inf ;
;



Parameter Result(iter,*);



Scalar Bounded;



Loop(iter$(NOT(Convergency)),

Bounded = YES;
Loop(Scen,

Prob_s=Prob(Scen);
b_s(n)=b(n,Scen);

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

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

uL(n,Iter,Scen) = u.l(n);

else

Bounded = NO;

Solve MSP us LP min Z_MSP;

uL(n,Iter,Scen) = u.l(n);
FC(Iter,Scen) = YES;
Result(iter,'FC')= YES;
*Result(iter,'MSP')= Z_MSP.l;

)
;

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

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

LB=  sum(scen, [sum(n,[b(n,scen) - sum(i,d(n,i)*yL(i))]*uL(n,Iter,Scen) )])
    + sum(i,f(i)*yL(i)) ;

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

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

yL(i) = y.l(i);
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
;
```



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**