

```
/*File will not run without the Constrained Optimization Module obtainable
at www.aptech.com*/
/*Settings for Constrained Optimization*/
new;
library co;
screen on;
coset;
format /M1 /ROS 16,5;
cls;
_co_Algorithm = 1;
_co_LineSearch = 2;
_co_Bounds = { 0 1 };
_co_MaxIters = 10000;
_co_DirTol = 3*1e-5;

/*Necessary Input: P= Vector of Interst Positions; R= Vector of Actors' Power;
Vector containing informaiton on the kind of actor 1= non-state, 2=state, 3=leading
ministry*/

P = {0.1, 0.4, 0.3, 0.5, 0.8, 1, 1};
R = {0.2, 0.2, 1, 1, 1, 0.3, 0.3 };
K = { 1,1,2,2,3,1,1};

/*Set a = 1 for asymmetric version*/
a= 0;
/*****/

/*Relative Power*/
R = R./sumc(R);

/*Defining the Disagreement Value D*/
D = zeros(rows(P),1);
i = 1;
do while i <= rows(P);
if P[i,1] > 0.5;
D[i,1] = (1-R[i,1])*P[i,1];
endif;
if P[i,1] <= 0.5;
D[i,1] = (1-R[i,1])*(1-P[i,1]);
endif;
i = i+1;
endo;

/*Constraining the possible outcome on pareto efficiency*/
proc dumm(x);
local nc;
nc = D-abs(P-x);
retp(minc(nc)-1e-5);
endp;

/*Setting the Starting Vector on the Mean Position*/
x0 = meanc(P);
```

```
/* Activation of CO-procedure and constraint*/
_co_IneqProc = &dumm;
{ x, f, g, retcode } = co(&fsys, x0) ;

/*Procedure containing the utility functions for NBS*/
if a==0;
R = 1+zeros(rows(R),1);
endif;
/*Relative Power*/
R = R./sumc(R);
proc fsys(x);
local uf, er;
uf = (D-abs(P-x[1])).^R;
er = prodc(uf);
retp(-1*er);
endp;
call coPRT(x, f,g,retcode);

/*Return predicted Outcome!*/
print x;
```

```

/*File will not run without the Constrained Optimization Module obtainable
at www.aptech.com*/
/*Settings for Constrained Optimization*/
new;
library co;
screen on;
coset;
format /M1 /ROS 16,5;
cls;
_co_Algorithm = 1;
_co_LineSearch = 2;
_co_Bounds = { 0 1 };
_co_MaxIters = 10000;
_co_DirTol = 3*1e-5;

/*Necessary Input: P= Vector of Interst Positions; R= Vector of Actors' Power;
Vector containing informaiton on the kind of actor 1= non-state, 2=state, 3=leading
ministry*/

P = {0.1, 0.4, 0.3, 0.5, 0.8, 1, 1};
R = {0.2, 0.2, 1, 1, 1, 0.3, 0.3 };
K = { 1,1,2,2,3,1,1};

/*Set a = 1 for asymmetric version*/
a= 0;
/*****/

/*Relative Power*/
R = R./sumc(R);

/*Defining the Disagreement Value D*/
D = zeros(rows(P),1);
i = 1;
do while i <= rows(P);
if P[i,1] > 0.5;
D[i,1] = (1-R[i,1])*P[i,1];
endif;
if P[i,1] <= 0.5;
D[i,1] = (1-R[i,1])*(1-P[i,1]);
endif;
i = i+1;
endo;

/*Constraining the possible outcome on pareto efficiency*/
proc dumm(x);
local nc;
nc = D-abs(P-x);
retp(minc(nc)-1e-5);
endp;

/*Setting the Starting Vector on the Mean Position*/
x0 = meanc(P);

```

```
/* Activation of CO-procedure and constraint*/
_co_IneqProc = &dumm;
{ x, f, g, retcode } = co(&fsys, x0) ;

/*Procedure containing the utility functions for NBS*/
if a==0;
R = 1+zeros(rows(R),1);
endif;
/*Relative Power*/
R = R./sumc(R);
proc fsys(x);
local uf, er;
uf = (D-abs(P-x[1])).^R;
er = prodc(uf);
retp(-1*er);
endp;
call coPRT(x, f,g,retcode);

/*Return predicted Outcome!*/
print x;
```

```

/*File will not run without the Constrained Optimization Module obtainable
at www.aptech.com*/
/*Settings for Constrained Optimization*/
new;
library co;
screen on;
coset;
format /M1 /ROS 16,5;
cls;
_co_Algorithm = 1;
_co_LineSearch = 2;
_co_Bounds = { 0 1 };
_co_MaxIters = 10000;
_co_DirTol = 3*1e-5;

/*Necessary Input: P= Vector of Interst Positions; R= Vector of Actors' Power;
Vector containing informaiton on the kind of actor 1= non-state, 2=state, 3=leading
ministry*/

P = {0.1, 0.4, 0.3, 0.5, 0.8, 1, 1};
R = {0.2, 0.2, 1, 1, 1, 0.3, 0.3 };
K = { 1,1,2,2,3,1,1};

/*Set a = 1 for asymmetric version*/
a= 0;
/*****/

/*Relative Power*/
R = R./sumc(R);

/*Defining the Disagreement Value D*/
D = zeros(rows(P),1);
i = 1;
do while i <= rows(P);
if P[i,1] > 0.5;
D[i,1] = (1-R[i,1])*P[i,1];
endif;
if P[i,1] <= 0.5;
D[i,1] = (1-R[i,1])*(1-P[i,1]);
endif;
i = i+1;
endo;

/*Constraining the possible outcome on pareto efficiency*/
proc dumm(x);
local nc;
nc = D-abs(P-x);
retp(minc(nc)-1e-5);
endp;

/*Setting the Starting Vector on the Mean Position*/
x0 = meanc(P);

```

```
/* Activation of CO-procedure and constraint*/
_co_IneqProc = &dumm;
{ x, f, g, retcode } = co(&fsys, x0) ;

/*Procedure containing the utility functions for NBS*/
if a==0;
R = 1+zeros(rows(R),1);
endif;
/*Relative Power*/
R = R./sumc(R);
proc fsys(x);
local uf, er;
uf = (D-abs(P-x[1])).^R;
er = prodc(uf);
retp(-1*er);
endp;
call coPRT(x, f,g,retcode);

/*Return predicted Outcome!*/
print x;
```