```
library("gurobi")
library("Matrix")

SP1 = function(u1, u2, ub=NULL, lb=NULL){
  mySP1 = list()
  mySP1$modelsense = "min"
  mySP1$obj = c(15+0.7+u1,10+0.7+u2,15+0.4+u1,10+0.4+u2)
  mySP1$A = Matrix(c(0.4,0.32,0.4,0.32,0.2,0.4,0.2,0.4), nrow=2, ncol=4, byrow=T, sparse=T)
  mySP1$sense = c(">=",">=")
  mySP1$rhs = c(300,800)
  mySP1$vtypes = "I"
  mySP1$ub = ub
  mySP1$lb = lb

  mysol = gurobi(mySP1)
  x.h1 = mysol$x
  z.SP1 = mysol$objval
  list(x.h1 = x.h1, z.SP1 = z.SP1)
}

SP2 = function(u1, u2, ub=NULL, lb=NULL){
  mySP2 = list()
  mySP2$modelsense = "min"
  mySP2$obj = c(15+0.5+u1,10+0.5+u2,15+0.9+u1,10+0.9+u2)
  mySP2$A = Matrix(c(0.4,0.32,0.4,0.32,0.2,0.4,0.2,0.4,0.35,0.2,0.35,0.2), nrow=3, ncol=4, byrow=T, sparse=T)
  mySP2$sense = c(">=",">=",">=")
  mySP2$rhs = c(600,400,800)
  mySP2$vtypes = "I"
  mySP2$ub = ub
  mySP2$lb = lb

  mysol = gurobi(mySP2)
  x.h2 = mysol$x
  z.SP2 = mysol$objval
  list(x.h2 = x.h2, z.SP2 = z.SP2)
}

SP3 = function(u1, u2, ub=NULL, lb=NULL){
  mySP3 = list()
  mySP3$modelsense = "min"
  mySP3$obj = c(15+0.7+u1,10+0.7+u2,15+0.4+u1,10+0.4+u2)
  mySP3$A = Matrix(c(0.4,0.32,0.4,0.32,0.2,0.4,0.2,0.4,0.35,0.2,0.35,0.2), nrow=3, ncol=4, byrow=T, sparse=T)
  mySP3$sense = c(">=",">=",">=")
  mySP3$rhs = c(900,300,500)
  mySP3$vtypes = "I"
  mySP3$ub = ub
  mySP3$lb = lb

  mysol = gurobi(mySP3)
  x.h3 = mysol$x
  z.SP3 = mysol$objval
  list(x.h3 = x.h3, z.SP3 = z.SP3)
}

MP = function(myMP, x.h){
  myMP$A = rBind(myMP$A, Matrix(c(1,0,0,-(x.h[1] + x.h[3]),-(x.h[2] + x.h[4])),nrow=1, ncol=5, byrow=T, sparse=T))
  myMP$sense = c(myMP$sense, "<=")
  myMP$rhs = c(myMP$rhs, 15.7*x.h[1] + 10.7*x.h[2] + 15.4*x.h[3] + 10.4*x.h[4])

  myMP$A = rBind(myMP$A, Matrix(c(0,1,0,-(x.h[5] + x.h[7]),-(x.h[6] + x.h[8])),nrow=1, ncol=5, byrow=T, sparse=T))
  myMP$sense = c(myMP$sense, "<=")
  myMP$rhs = c(myMP$rhs, 15.5*x.h[5] + 10.5*x.h[6] + 15.9*x.h[7] + 10.9*x.h[8])

  myMP$A = rBind(myMP$A, Matrix(c(0,0,1,-(x.h[9] + x.h[11]),-(x.h[10] + x.h[12])),nrow=1, ncol=5, byrow=T, sparse=T))
  myMP$sense = c(myMP$sense, "<=")
  myMP$rhs = c(myMP$rhs, 15.7*x.h[9] + 10.7*x.h[10] + 15.4*x.h[11] + 10.4*x.h[12])

  mysol = gurobi(myMP)

  theta1 = mysol$x[1]
  theta2 = mysol$x[2]
  theta3 = mysol$x[3]
  u1 = mysol$x[4]
  u2 = mysol$x[5]
  alpha = mysol$pi

  list(u1=u1, u2=u2, alpha=alpha, UB = mysol$objval, myMP = myMP)
```

```
}


### Iteration 0: Initialize ###

UB = Inf
LB = -Inf
Incumbent = -Inf
X = NULL


### Subproblems ###
u1 = 100
u2 = 100

mySP3 = SP3(u1, u2)
mySP1 = SP1(u1, u2)
mySP2 = SP2(u1, u2)

x.h = c(mySP1$x.h1, mySP2$x.h2, mySP3$x.h3)
X = cBind(X, x.h)
LB = max(LB, (mySP1$z.SP1 + mySP2$z.SP2 + mySP3$z.SP3 - 4000*u1 - 3000*u2))

### Master Problem ###

myMP = list()
myMP$modelsense = "max"
myMP$obj = c(1,1,1,-4000,-3000)
myMP$A = Matrix(c(1,0,0,-(x.h[1] + x.h[3]),-(x.h[2] + x.h[4]),
                  0,1,0,-(x.h[5] + x.h[7]),-(x.h[6] + x.h[8]),
                  0,0,1,-(x.h[9] + x.h[11]),-(x.h[10] + x.h[12])),
                nrow=3, ncol=5, byrow=T, sparse=T)
myMP$sense = c("<=","<=","<=")
myMP$rhs = c(15.7*x.h[1] + 10.7*x.h[2] + 15.4*x.h[3] + 10.4*x.h[4],
             15.5*x.h[5] + 10.5*x.h[6] + 15.9*x.h[7] + 10.9*x.h[8],
             15.7*x.h[9] + 10.7*x.h[10] + 15.4*x.h[11] + 10.4*x.h[12])

myMP$vtypes = "C"
myMP$lb = c(-1000000, -1000000, -1000000, 0, 0)
myMP$ub = c(1000000, 1000000, 1000000, 1000000, 1000000)

mysol = gurobi(myMP)

theta1 = mysol$x[1]
theta2 = mysol$x[2]
theta3 = mysol$x[3]
u1 = mysol$x[4]
u2 = mysol$x[5]

alpha = mysol$pi

UB = mysol$objval

x = x.h

check = (LB==UB)

while(!check){
  mySP1 = SP1(u1, u2)
  mySP2 = SP2(u1, u2)
  mySP3 = SP3(u1, u2)

  x.h = c(mySP1$x.h1, mySP2$x.h2, mySP3$x.h3)
  X = cBind(X, x.h)
  LB = max(LB, (mySP1$z.SP1 + mySP2$z.SP2 + mySP3$z.SP3 - 4000*u1 - 3000*u2))

  MP.out = MP(myMP, x.h)
  myMP = MP.out$myMP
  alpha = MP.out$alpha
  u1 = MP.out$u1
  u2 = MP.out$u2
  UB = MP.out$UB

  x.SP1 = X[1:4,] %*% alpha[seq(1, length(alpha), 3)]
  x.SP2 = X[5:8,] %*% alpha[seq(2, length(alpha), 3)]
  x.SP3 = X[9:12,] %*% alpha[seq(3, length(alpha), 3)]

  x = rbind(x.SP1, x.SP2, x.SP3)
```

```
  check = (LB==UB)
}
LB
UB
```