31761 - Renewables in Electricity Markets

Implementing an LP-based market-clearing in R

The aim of this document is to describe how to implement the most basic version of Linear Program (LP) for market-clearing, as well as its dual version to obtain the equilibrium price. The example considered in the following is a direct implementation of that in the slides of "Lecture 3: Day-ahead markets". Note that the package lpSolve needs to be installed since needed as the LP solver here. It is available, as well as necessary documentation, at teh following link.

Case 1: Implementation of the primal LP for market clearing

This example implementation directly relates to slides 14-17 in "Lecture 3: Day-ahead markets". This function can be saved into a file called marketclearing.R. Then in R, one may start by loading this source file,

```
> source('marketclearing.R')
```

before to simply call that the function marketclearingLP,

```
> marketclearingLP()
```

That function is extensively described and documented below. Note that one can directly the equilibrium price from lpSolve by asking for the dual variable related to the balance equality constraint (at the very end of the function).

```
marketclearingLP <- function(){

require(lpSolve) # calls the necessary LP solver (to be installed beforehand!)

# Declare the list of demand offers
xd <- c(250,300,120,80,40,70,60,45,30,35,25,10) # Quantities
yd <- c(200,110,100,90,85,75,65,40,37.5,30,24,15) # Prices

# Declare the list of supply offers
xg <- c(120,50,200,400,60,50,60,100,70,50,70,45,50,60,50) # Quantities
yg <- c(0,0,15,30,32.5,34,36,37.5,39,40,60,70,100,150,200) # Prices

# Set up the minimization problem as an LP to be solved by lpSolve...

# Construct the c vector for the objective function:
f.obj <- c(yg,-yd)

# Construct the A_eq vector for the equality constraint
eq.c  <- c(rep(1,length(yg)),rep(-1,length(yd)))

# Construct the A matrix for the inequality constraints (as well as
# non-negativity)
oneg  <- diag(rep(1,length(xg)))
zerod <- array(0,dim=c(length(xg),length(xd)))
oned  <- diag(rep(1,length(xd)))
zerog <- array(0,dim=c(length(xd),length(xg)))
mgd   <- cbind(oneg,zerod)
mgd2  <- rbind(mgd,mgd)
mdg   <- cbind(zerog,oned)
mdg2  <- rbind(mdg,mdg)
```

```
Bmdg  <- rbind(mgd2,mdg2)

# Bind A_eq and A together
f.con <- rbind(eq.c, Bmdg)

# Construct the right-hand side of the equality constraint
beq <- 0

# Construct the right-hand side of the inequality constraint
b <- c(rep(0,length(yg)),xg,rep(0,length(yd)),xd)

# Bind b_eq and b together
f.rhs <- c(beq,b)

# Declare the all types of constraints to be represented (i.e., 1 equality
# constraint, and a suite of inequality constraints)
f.dir <- c("==",rep(">=",length(yg)),rep("<=",length(yg)),
rep(">=",length(yd)),rep("<=",length(yd)))

# Now run feed the LP solver
lp ("min", f.obj, f.con, f.dir, f.rhs)

# ... and ask for the solution
v.sol <- lp ("min", f.obj, f.con, f.dir, f.rhs)$solution

# The solution can then be decomposed into:
# 1. the dispatch for the suppliers
print(c("Supply: ",v.sol[1:length(yg)]))
# 2. the dispatch for the demand
print(c("Demand: ",v.sol[(length(yg)+1):(length(yg)+length(yd))]))

# And one can even directly get the equilibrium price as the dual
# variable for the balance constraint
eq.price <- lp ("min", f.obj, f.con, f.dir, f.rhs,compute.sens=TRUE)$duals[1]
print(c("Equilibrium price: ",eq.price))
}
```

Case 2: Implementation of the dual LP

This example implementation directly relates to slides 19-22 in "Lecture 3: Day-ahead markets". As for the above, this function can also be saved into the same file called marketclearing.R. Then in R, by first loading this source file,

```
> source('marketclearing.R')
```

one may then call the function marketclearingdualLP,

```
> marketclearingdualLP()
```

That function is extensively described and documented below.

```
marketclearingdualLP <- function(){

require(lpSolve) # calls the necessary LP solver (to be installed beforehand!)

# Declare the list of demand offers
xd <- c(250,300,120,80,40,70,60,45,30,35,25,10) # Quantities
yd <- c(200,110,100,90,85,75,65,40,37.5,30,24,15) # Prices
```

```
# Declare the list of supply offers
xg <- c(120,50,200,400,60,50,60,100,70,50,70,45,50,60,50) # Quantities
yg <- c(0,0,15,30,32.5,34,36,37.5,39,40,60,70,100,150,200) # Prices

# Set up the minimization problem as an LP to be solved by lpSolve...

# Construct the c vector for the objective function:
f.obj <- c(-xg,-xd,0)

# Construct the A matrix for the inequality constraints (as well as
# non-negativity)
onec   <- diag(rep(-1,length(xg)+length(xd)))
rcol   <- c(rep(1,length(xg)),rep(-1,length(xd)))
mgd    <- cbind(onec,rcol)
onecm  <- diag(rep(1,length(xg)+length(xd)))
rcolm  <- rep(0,length(xg)+length(xd))
mdg    <- cbind(onecm,rcolm)
f.con  <- rbind(mgd,mdg)

# Construct the right-hand side of the inequality constraint
f.rhs <- c(yg,-yd,rep(0,length(xg)+length(xd)))

# Declare the all types of constraints to be represented (i.e.,
# a suite of inequality constraints)
f.dir <- c(rep("<=",(length(xg)+length(xd))),rep(">=",(length(xg)+length(xd))))

# Now feed the LP solver, and directly ask for the solution
v.sol <- lp ("max", f.obj, f.con, f.dir, f.rhs)$solution
# The equilibrium price is related to the last constraint
eq.price <- v.sol[length(v.sol)]
print(c("Equilibrium price: ",eq.price))
}
```