```
# Autor: Marcos Daniel Baroni
# Created: Mon Jan 21 10:39:57 BRST 2014
# Updated: Mon Jan 27 16:09:43 BRST 2014
#
#####################
# Decision Variable #
#####################

# Number of actions by given period
var x[Acs*Pers] integer;


####################
# Others variables #
####################

# Energy recover for a given period caused by
a given action
#   rec[a, k1, k2] is the energy recovered by
action "a" (taken made on period "k1"),
#   "k2" after it
var rec[Acs*DPers];

# Profit for energy recovering for a given per
iod
var prof[DPers];

# Total cost of all actions executed on a give
n period
var cost[Pers];


#############
# Equations #
#############
# Total energy recovered on Period "k" by acti
on "i"
subto rec_def:
  forall <i, k> in Acs*DPers do
        sum <k2> in Pers with (k-k2+1) > 0 and
 (k-k2+1) <= Y*P do
               x[i, (k-k2+1)]*e[i, k2] == rec
[i, k];

# Profit by energy recovery for period k>
subto prof_def:
  forall <k> in DPers do
    sum <i> in Acs do
          rec[i, k]*v[i] == prof[k];

# Cost of all actions on period K
subto cost_def:
  forall <k> in Pers do
    sum <i, l> in Acs*Res do
          x[i, k]*c[i, l] == cost[k];

###############
# Constraints #
###############

# Anual Goal
subto anual_goal:
  forall <j> in Yrs do
    sum <i, k> in Acs*YPers[j] do
          rec[i, k] <= g[j];

# Global Budgets
#subto global_budget:
#  forall <l> in Res do
#    sum <i, k> in Acs*Pers do
#         x[i, k]*c[i, l] <= o[l];

# Anual Budgets
subto anual_budget:
  forall <j, l> in Yrs*Res do
    sum <i, k> in Acs*YPers[j] do
          x[i, k]*c[i, l] <= p[j, l];

# PERIODIC Budgets
#subto PERIODIC_budget:
#  forall <k, l> in Pers*Res do
#    sum <i> in Acs do
#         x[i, k] <= s[l, k];

# Global Market
subto global_market:
  forall <i> in Acs do
    sum <k> in Pers do
          x[i, k] <= m[i];

# Anual Market
subto anual_market:
  forall <i, j> in Acs*Yrs do
    sum <k> in YPers[j] do
          x[i, k] <= u[i, j];

# PERIODIC Market
subto PERIODIC_market:
  forall <i, k> in Acs*Pers do
          x[i, k] <= z[i, k];

# Dependecy between actions
subto dependency:
        forall <i1, i2, q> in D do
                forall <k> in Pers do
                        sum <k2> in Pers with
(k2 < k) do
                                x[i1, k2] <=
                        sum <k3> in Pers with
(k3 < k) do
                                q*x[i2, k3];


######################
# Objective Function #
######################
maximize npv:
        sum <k> in DPers do
                prof[k]/(1+r)^k -
        sum <k> in Pers do
                cost[k]/(1+r)^k;
```