

```
# 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 #
#####

# (Rec) rec[i, j]: Energy recovered by actions
# 'i' on the jth-period of the
# plan
var rec[Acs*Pers];

# (Rec') rec2[i, j]: Energy recovered by actions
# 'i' on the jth-period AFTER the
# plan
var rec2[Acs*Pers];

# Total cost of all actions executed on a given
# period
var cost[Acs*Pers];

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

# Total energy recovered on the "k"-th period
# after plan, by action "i"
subto rec_def2:
    forall <i, k> in Acs*Pers do
        sum <k2> in Pers with k2 >= k+1 do
            x[i, k2]*e[i, (Y*P+k-k2+1)] ==
rec2[i, k];

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

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

# Annual 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];

# Annual 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];

# Annual 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];

# Dependency 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 <i> in Acs do
        sum <k> in Pers do
            (rec[i, k]*v[i] - cost
[i, k])/((1+r)^k)
        +
        sum <i> in Acs do
            sum <k> in Pers do
                rec2[i, k]*v[i]/((1+r)
^(Y*P+k));
```

