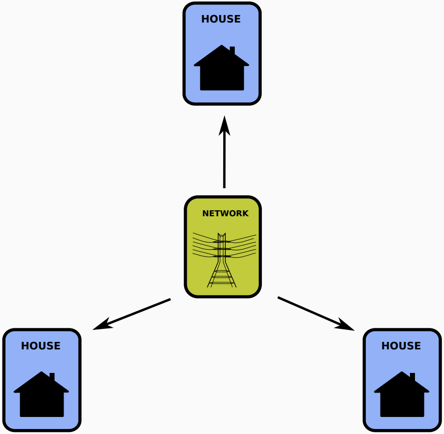
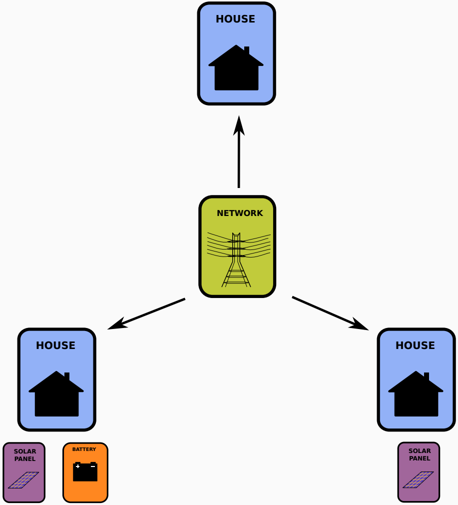
Optimization of an urban district microgrid

F. Pacaud
Advisors: P. Carpentier, J.-P. Chancelier, M. De Lara
November 9, 2016

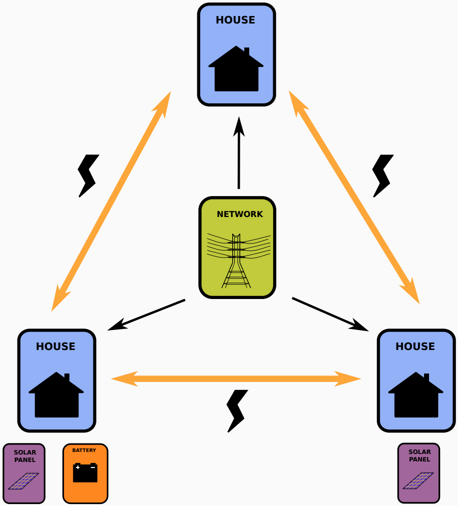
Usually houses import electricity from the grid



But more and more houses are equipped with solar panel



Is it worth to add a local grid to exchange electricity?



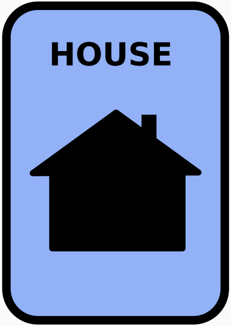
Is it worth to connect different houses together inside a district?

Challenges:

- Handle electrical exchanges between houses

We turn to mathematical optimization to answer the question

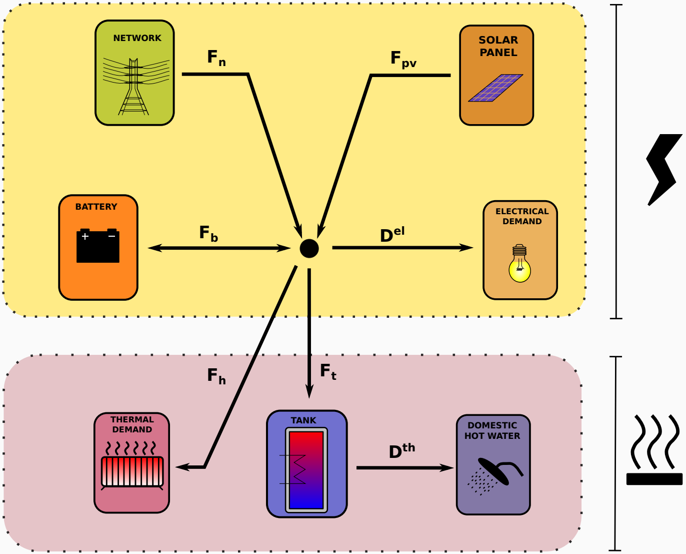
Two commandments to rule them all



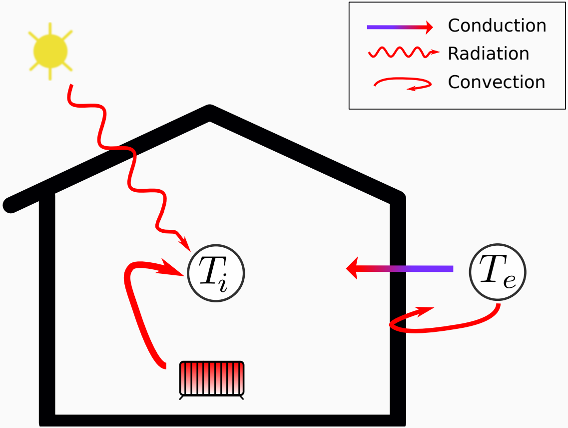
Thou shall:

- Satisfy thermal comfort
- Optimize operational costs

For each house, we consider the electrical system...

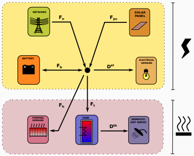


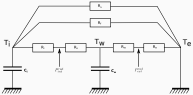
… and the thermal enveloppe



Outline

We introduce states, controls and noises

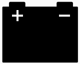




- **Stock variables** $X_t = \big(B_t, H_t, \theta^i_t, \theta^w_t\big)$
  - B_t, battery level (kWh)
  - H_t, hot water storage (kWh)
  - θ^i_t, inner temperature ($^\circ C$)
  - θ^w_t, wall's temperature ($^\circ C$)
- **Control variables** $U_t = \big(F^+_{B,t}, F^-_{B,t}, F_{T,t}, F_{H,t}\big)$
  - $F^+_{B,t}$, energy stored in the battery
  - $F^-_{B,t}$, energy taken from the battery
  - $F_{T,t}$, energy used to heat the hot water tank
  - $F_{H,t}$, thermal heating
- **Uncertainties** $W_t = \big(D^E_t, D^{DHW}_t, P^{ext}_t, \theta^e_t\big)$
  - D^E_t, electrical demand (kW)
  - D^{DHW}_t, domestic hot water demand (kW)
  - P^{ext}_t, external radiations (kW)
  - θ^e_t, external temperature ($^\circ C$)

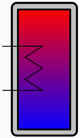
Discrete time state equations

So we have the four state equations (all linear):



$$B_{t+1} = \alpha_B B_t + \Delta T \left(\rho_c F^{+}_{B,t} - \frac{1}{\rho_d} F^{-}_{B,t} \right)$$

$$H_{t+1} = \alpha_H H_t + \Delta T \left[F_{T,t} - D^{DHW}_t \right]$$



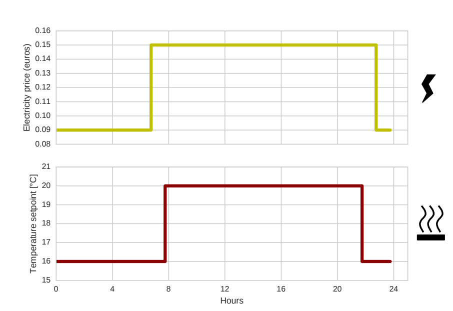
$$\theta^w_{t+1} = \theta^w_t + \frac{\Delta T}{c_m} \left[\frac{\theta^i_t - \theta^w_t}{R_i + R_s} + \frac{\theta^e_t - \theta^w_t}{R_m + R_e} + \gamma F_{H,t} + \frac{R_i}{R_i + R_s} P^{int}_t + \frac{R_e}{R_e + R_m} P^{ext}_t \right]$$

$$\theta^i_{t+1} = \theta^i_t + \frac{\Delta T}{c_i} \left[\frac{\theta^w_t - \theta^i_t}{R_i + R_s} + \frac{\theta^e_t - \theta^i_t}{R_v} + \frac{\theta^e_t - \theta^i_t}{R_f} + (1 - \gamma) F_{H,t} + \frac{R_s}{R_i + R_s} P^{int}_t \right]$$

which will be denoted:

$$\boxed{X_{t+1} = f_t(X_t, U_t, W_{t+1})}$$

Prices and temperature setpoints vary along time



- $T_f = 24\text{h}$, $\Delta T = 15\text{mn}$
- Electricity peak and off-peak hours
- $\pi_t^E = 0.09$ or 0.15 euros/kWh
- Temperature set-point $\bar{\theta}_t^i = 16^\circ C$ or $20^\circ C$

The costs we have to pay

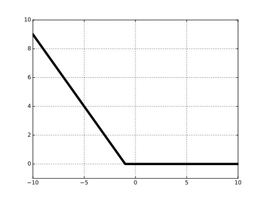
- Cost to import electricity from the network

$$-\underbrace{b_t^E \max\{0, -F_{NE,t+1}\}}_{\text{selling}} + \underbrace{\pi_t^E \max\{0, F_{NE,t+1}\}}_{\text{buying}}$$

where we define the recourse variable (electricity balance):

$$\underbrace{F_{NE,t+1}}_{Network} = \underbrace{D^E_{t+1}}_{Demand} + \underbrace{F^+_{B,t} - F^-_{B,t}}_{Battery} + \underbrace{F_{H,t}}_{Heating} + \underbrace{F_{T,t}}_{Tank} - \underbrace{F_{pv,t}}_{Solar\ \ panel}$$

- Virtual Cost of thermal discomfort: $\kappa_{th}(\underbrace{\theta^i_t - \bar{\theta}^i_t}_{\text{deviation from setpoint}})$



κ_{th}
Piecewise linear cost
Penalize temperature if
below given setpoint

Instantaneous and final costs for a single house

- The instantaneous convex costs are

$$L_t(X_t, U_t, W_{t+1}) = \underbrace{-b_t^E \max\{0, -F_{NE,t+1}\}}_{\textit{buying}} + \underbrace{\pi_t^E \max\{0, F_{NE,t+1}\}}_{\textit{selling}} + \underbrace{\kappa_{th}(\theta_t^i - \bar{\theta}_t^i)}_{\textit{discomfort}}$$

- We add a final linear cost

$$K(X_{T_f}) = -\pi^H H_{T_f} - \pi^B B_{T_f}$$

to avoid empty stocks at the final horizon T_f

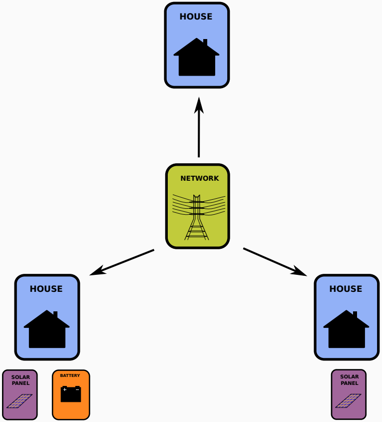
That gives the following stochastic optimization problem

$$\min_{X,U} \quad J(X,U) = \mathbb{E}\left[\sum_{t=0}^{T_f-1} \underbrace{L_t(X_t, U_t, W_{t+1})}_{instantaneous\ \ cost} + \underbrace{K(X_{T_f})}_{final\ \ cost}\right]$$

$$\begin{aligned} s.t \quad X_{t+1} = f_t(X_t, U_t, W_{t+1}) \quad & \text{Dynamic} \\ X^\flat \leq X_t \leq X^\sharp \quad & \\ U^\flat \leq U_t \leq U^\sharp \quad & \\ X_0 = X_{ini} \quad & \\ \sigma(U_t) \subset \sigma(W_1, \ldots, W_t) \quad & \text{Non-anticipativity} \end{aligned}$$

Outline

We have three different houses



Our (small) district:

- House 1: solar panel + battery
- House 2: solar panel
- House 3: nothing

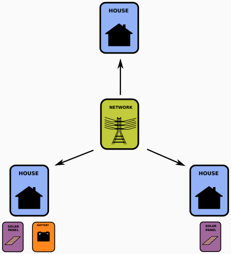
For the three houses:

- 10 stocks (= 4 + 3 + 3)
- 8 controls (= 4 + 2 + 2)
- 8 uncertainties
  (2 uncertainties in common)

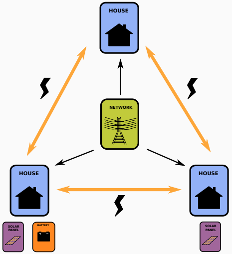
The total demand to the network is bounded:

$$\sum_{k=1}^{3} F^{k}_{NE,t+1} \leq F^{\sharp}_{NE}$$

We want to compare two configurations



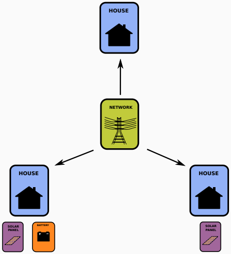
No exchange between houses



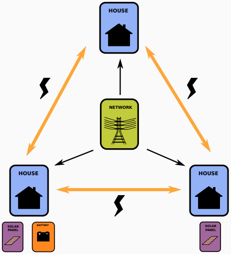
Exchange in a local grid

How much costs decrease
while allowing houses to exchange energy
through a local grid?

We want to compare two configurations



No exchange between houses

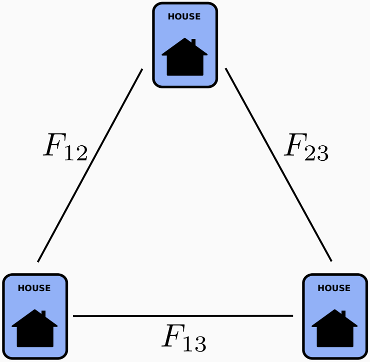


Exchange in a local grid

How much costs decrease
while allowing houses to exchange energy
through a local grid?

We show that local grid + optimization
decreases costs by 23 % during summer!

The grid adds three controls to the problem



How to solve this stochastic optimal control problem?

We recall the different parameters of our multistage stochastic problem:

- 96 timesteps ($= 4$ x 24)
- 10 stocks
- 11 controls
- 8 uncertainties

How to solve this stochastic optimal control problem?

We recall the different parameters of our multistage stochastic problem:

- 96 timesteps (= 4 x 24)
- 10 stocks
- 11 controls
- 8 uncertainties

The state dimension is high (=10), the problem is not tractable by a straightforward use of *dynamic programming* because of the curse of dimensionality! :-(

How to solve this stochastic optimal control problem?

We recall the different parameters of our multistage stochastic problem:

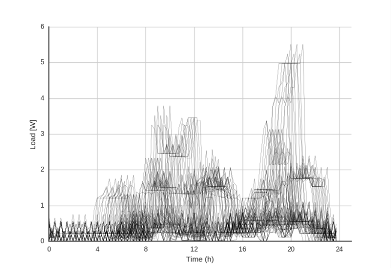
- 96 timesteps (= 4 x 24)
- 10 stocks
- 11 controls
- 8 uncertainties

The state dimension is high (=10), the problem is not tractable by a straightforward use of *dynamic programming* because of the curse of dimensionality! :-(

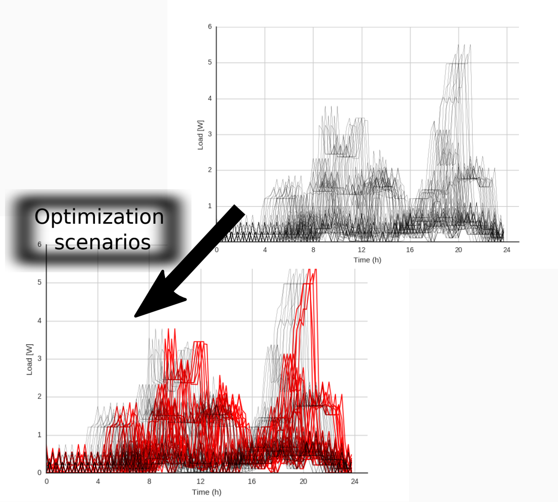
We will compare two methods that overcome this curse:

1. **Model Predictive Control** (MPC)
2. **Stochastic Dual Dynamic Programming** (SDDP)

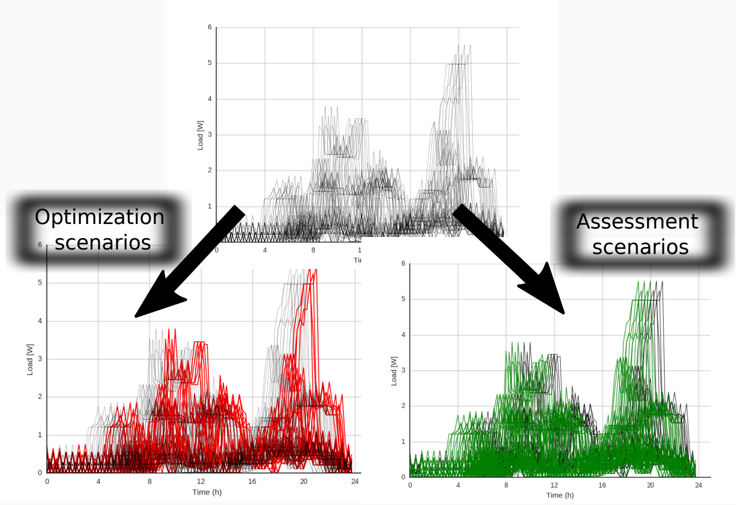
Out-of-sample comparison



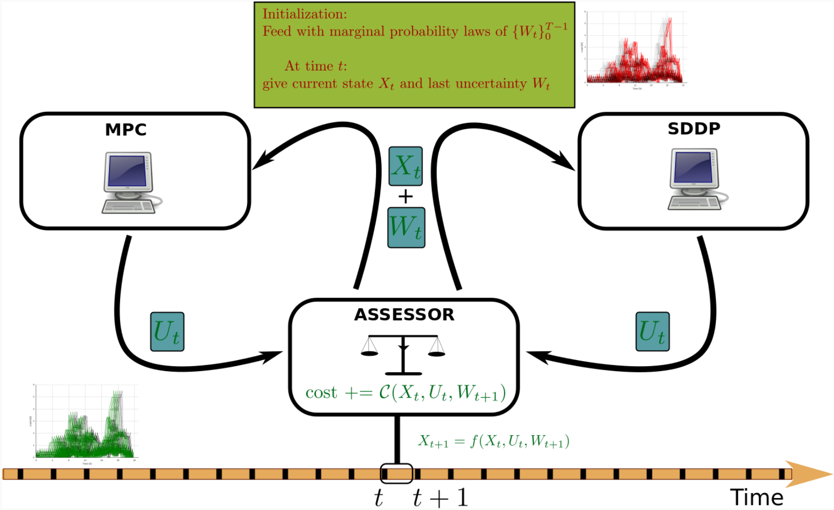
Out-of-sample comparison



Out-of-sample comparison



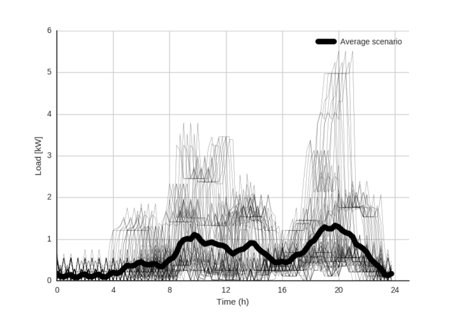
We compare SDDP and MPC with assessment scenarios



MPC vs SDDP: uncertainties modelling

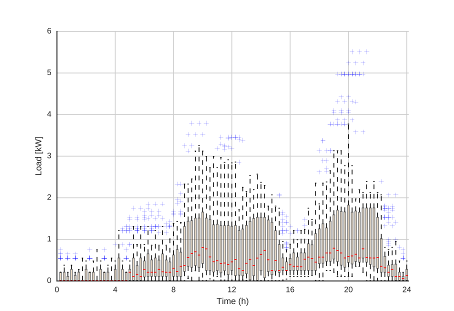
The two algorithms use optimization scenarios to model the uncertainties:

MPC



MPC considers the average. . .

SDDP



. . . and SDDP discrete laws

MPC vs SDDP: online resolution

At the beginning of time period $[\tau, \tau+1]$, do

MPC

- Consider a **rolling horizon** $[\tau, \tau+H[$
- Consider a **deterministic scenario** of demands (forecast) $(\overline{W}_{\tau+1}, \ldots, \overline{W}_{\tau+H})$
- Solve the **deterministic optimization** problem

$$\min_{X,U} \left[\sum_{t=\tau}^{\tau+H} L_t(X_t, U_t, \overline{W}_{t+1}) + K(X_{\tau+H}) \right]$$

$$\begin{aligned} s.t. \quad X. &= (X_\tau, \ldots, X_{\tau+H}) \\ U. &= (U_\tau, \ldots, U_{\tau+H-1}) \\ X_{t+1} &= f(X_t, U_t, \overline{W}_{t+1}) \\ X^\flat &\leq X_t \leq X^\sharp \\ U^\flat &\leq U_t \leq U^\sharp \end{aligned}$$

- Get optimal solution $(U^\#_\tau, \ldots, U^\#_{\tau+H})$ over horizon $H = 24h$
- Send only first control $U^\#_\tau$ to assessor, and iterate at time $\tau + 1$

SDDP

- We consider the approximated value functions $\left(\widetilde{V}_t\right)_0^{T_f}$

$$\underbrace{\widetilde{V}_t}_{\text{Piecewise affine functions}} \leq V_t$$

- Solve the **stochastic optimization problem**:

$$\min_{u_\tau} \mathbb{E}_{W_{\tau+1}} \Big[L_\tau(X_\tau, u_\tau, W_{\tau+1}) + \widetilde{V}_{\tau+1}\Big(f_\tau(X_\tau, u_\tau, W_{\tau+1})\Big)\Big]$$

$\Rightarrow$ this problem resumes to solve a LP at each timestep

- Get optimal solution $U^\#_\tau$
- Send $U^\#_\tau$ to assessor

A brief recall on Dynamic Programming

Dynamic Programming

μ_t is the probability law of W_t and is being used to estimate expectation and compute **offline** value functions with the backward equation:

$$V_T(x) = K(x)$$

$$V_t(x_t) = \min_{U_t} \mathbb{E}_{\mu_t}\Big[\underbrace{L_t(x_t, U_t, W_{t+1})}_{\text{current cost}} + \underbrace{V_{t+1}\Big(f(x_t, U_t, W_{t+1})\Big)}_{\text{future costs}}\Big]$$

A brief recall on Dynamic Programming

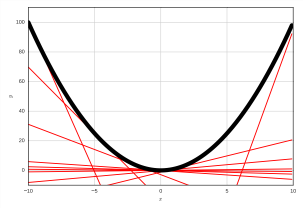
Dynamic Programming

μ_t is the probability law of W_t and is being used to estimate expectation and compute **offline** value functions with the backward equation:

$$V_T(x) = K(x)$$

$$V_t(x_t) = \min_{U_t} \mathbb{E}_{\mu_t}\Big[\underbrace{L_t(x_t, U_t, W_{t+1})}_{\text{current cost}} + \underbrace{V_{t+1}\Big(f(x_t, U_t, W_{t+1})\Big)}_{\text{future costs}}\Big]$$

Stochastic Dual Dynamic Programming



- Convex value functions V_t are approximated as a supremum of a finite set of affine functions
- Affine functions (=cuts) are computed during forward/backward passes, till convergence
- SDDP makes an extensive use of LP solver

$$\widetilde{V}_t(x) = \max_{1 \le k \le K} \{\lambda_t^k x + \beta_t^k\} \le V_t(x)$$

Outline

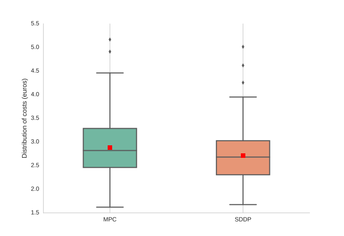
Our stack is deeply rooted in Julia language

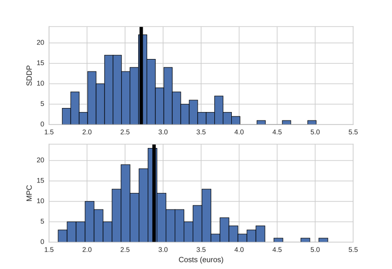
- Modeling Language: `JuMP`
- Open-source SDDP Solver: `StochDynamicProgramming.jl`
- LP Solver: `CPLEX 12.5`

`https://github.com/JuliaOpt/StochDynamicProgramming.jl`

Comparison of MPC and SDDP

We compare MPC and SDDP during one day in summer over 200 assessment scenarios:





	euros/day
MPC	2.882
SDDP	2.713

SDDP is in average *6.9 %* better than MPC!

Operational costs obtained in simulation

We compare different configurations, during summer and winter:

Summer

Local Grid	Elec. bill euros/day	Self cons. %
No	3.53	48.1 %
Yes	2.71	55.2 %

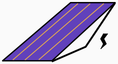
Winter

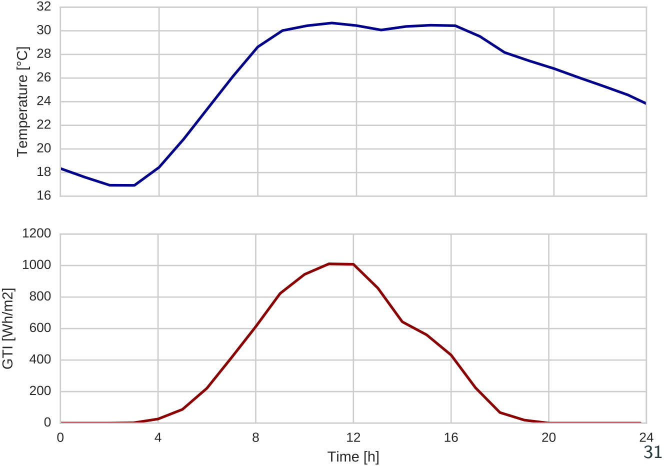
Local Grid	Elec. bill euros/day	Self cons. %
No	54.2	1.7 %
Yes	id.	id.

INPUT

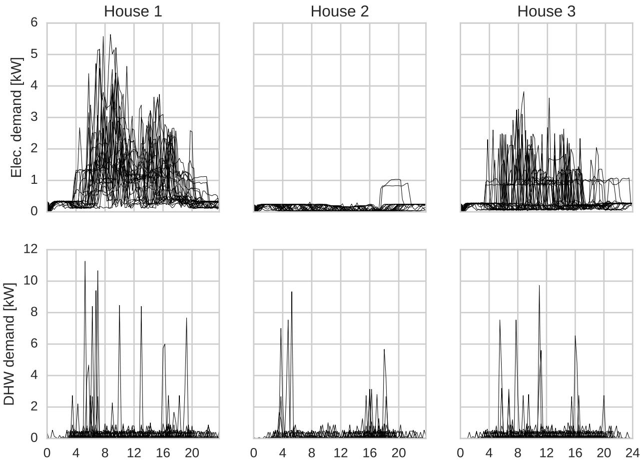
We work with real data

We consider one day during summer 2015 (data from Meteo France):





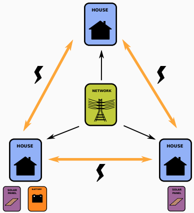
We have 200 scenarios of demands during this day

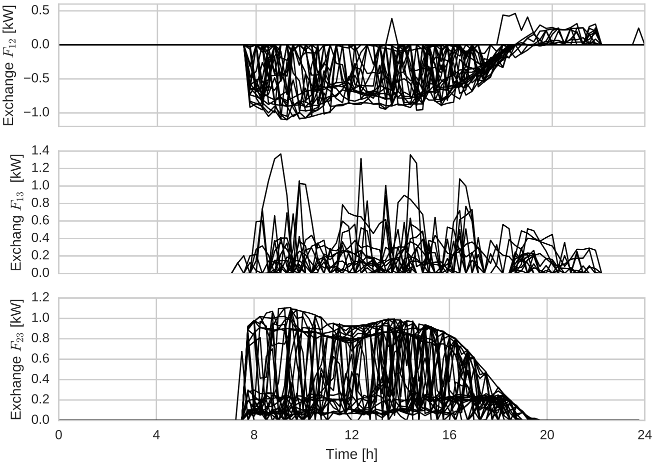


These scenarios are generated with `StRoBE`, a generator open-sourced by KU-Leuven

OUTPUT

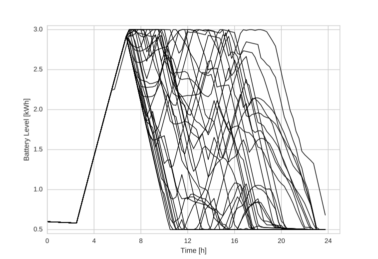
As we gain solar energy, surplus is traded in local grid



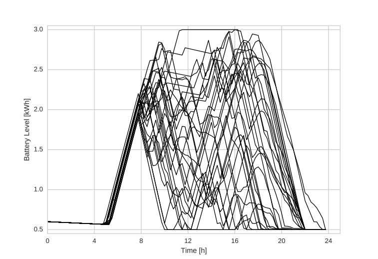


The battery is used as a global storage inside the local grid...

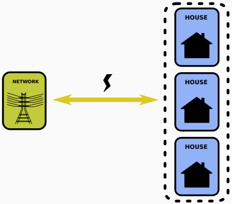
No local grid



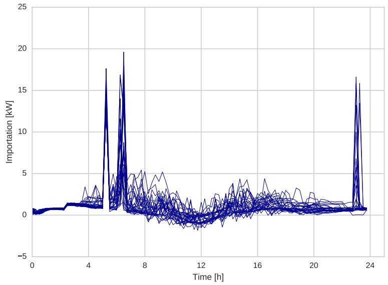
Local grid



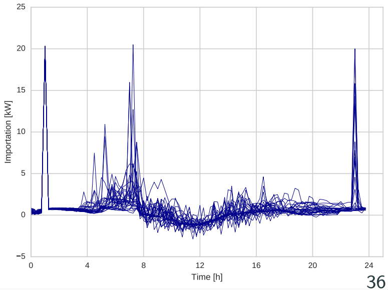
… and we minimize our average importation from the network



No local grid = 25.8 kWh



Local grid = 19.4 kWh



Outline

Conclusion

- We extend the results obtained with a single house to a small district
- This study can help to perform an economic analysis
- It pays to use stochastic optimization: SDDP is better than MPC
- We obtain promising results with SDDP, now we want to scale!

Perspectives

Mix SDDP with spatial decomposition like *Dual Approximate Dynamic Programming* (DADP) to control bigger urban neighbourhood