Comparison of MPC and SDDP to manage an urban district

Towards stochastic decomposition

F. Pacaud

Advisors: P. Carpentier, J.-P. Chancelier, M. De Lara

August 29, 2017

A paradigm shift in energy transition

The ambition of Efficacity is to improve urban energy efficiency.

Une loi encourage l'autoconsommation d'électricité

Jean-Claude Bourbon, le 17/02/2017 à 10h14
Mis à jour le 17/02/2017 à 10h31

Les professionnels n'ont pas attendu la fixation du cadre réglementaire pour lancer des offres.

De nombreuses jeunes sociétés investissent le créneau.

ZOOM

Un projet de loi vise à développer l'autoconsommation d'électricité. / diyanadimitrova/Fotolia

Le texte était réclamé depuis longtemps par les professionnels des én
renouvelables, en particulier dans le photovoltaïque. Le Parlement a

Self-consumption

Domestic storage

Energy management system

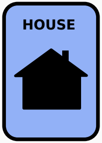
Our team focus on the control of *energy management system*.

What do we do

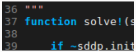
How to control storage inside urban microgrid ?

We follow a common procedure in operation research:

1. We consider a real world problem
   How to control a bunch of stocks ?

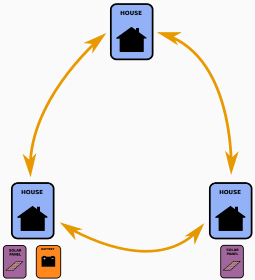


2. We model it as an optimization problem
   As demands are not predictable, we formulate a stochastic optimization problem



3. We develop algorithms to solve this particular optimization problem
   Dynamic Programming based methods, Model Predictive Control, ...

Analyzing the real world problem



We consider a system where different units (houses) are connected together via a local network (microgrid).

The houses have different stocks available:

- batteries,
- electrical hot water tank

and are equipped with solar panels.

We control the stocks every 15mn and we want to

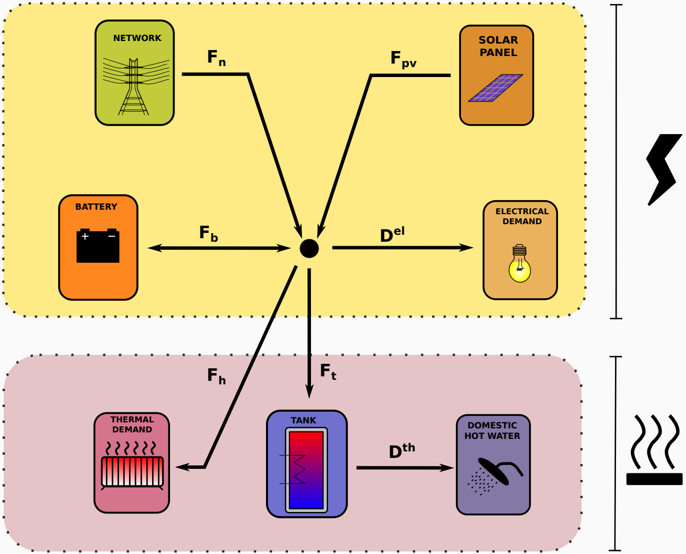
- minimize electric bill
- maintain a comfortable temperature inside the house

Outline

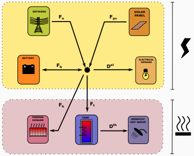
Physical modeling

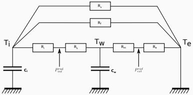
Outline

For each house, we consider the following devices



We introduce states, controls and noises

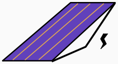


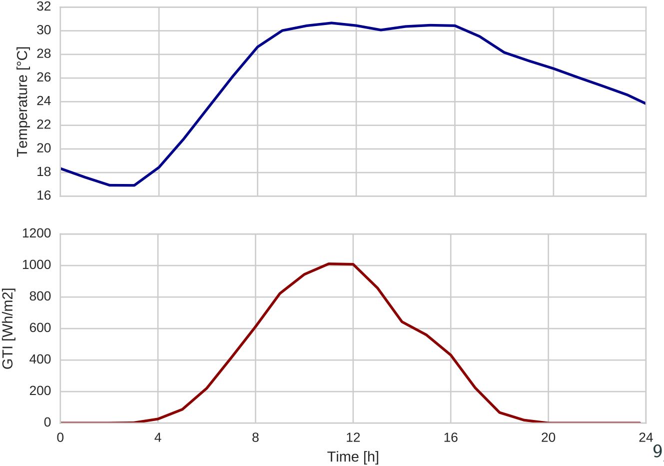


- **Stock variables** $\mathbf{X}_t = \big(\mathbf{B}_t, \mathbf{H}_t, \theta^i_t, \theta^w_t\big)$
  - $\mathbf{B}_t$, battery level (kWh)
  - $\mathbf{H}_t$, hot water storage (kWh)
  - θ^i_t, inner temperature ($^\circ C$)
  - θ^w_t, wall's temperature ($^\circ C$)
- **Control variables** $\mathbf{U}_t = \big(\mathbf{F}_{\mathbf{B},t}, \mathbf{F}_{T,t}, \mathbf{F}_{\mathbf{H},t}\big)$
  - $\mathbf{F}_{\mathbf{B},t}$, energy exchange with the battery (kW)
  - $\mathbf{F}_{T,t}$, energy used to heat the hot water tank (kW)
  - $\mathbf{F}_{\mathbf{H},t}$, thermal heating (kW)
- **Uncertainties** $\mathbf{W}_t = \big(\mathbf{D}^E_t, \mathbf{D}^{DHW}_t\big)$
  - $\mathbf{D}^E_t$, electrical demand (kW)
  - $\mathbf{D}^{DHW}_t$, domestic hot water demand (kW)

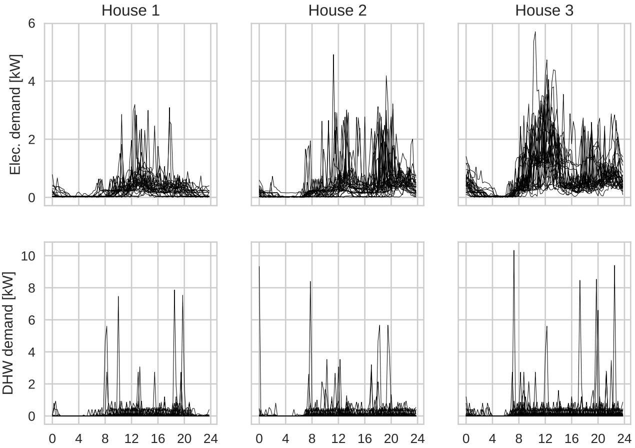
We work with real data

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





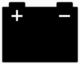
We generate scenarios of demands during this day



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

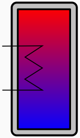
Discrete time state equations for each house

We have the four state equations (all linear), describing the stocks' evolution over time:



$$\mathbf{B}_{t+1} = \alpha_{\mathbf{B}} \mathbf{B}_t + \Delta T \Big(\rho_c \mathbf{F}^{+}_{\mathbf{B},t} - \frac{1}{\rho_d} \mathbf{F}^{-}_{\mathbf{B},t} \Big)$$

$$\mathbf{H}_{t+1} = \alpha_{\mathbf{H}} \mathbf{H}_t + \Delta T \big[\mathbf{F}_{T,t} - \mathbf{D}^{DHW}_t \big]$$



$$\boldsymbol{\theta}^{w}_{t+1} = \boldsymbol{\theta}^{w}_t + \frac{\Delta T}{c_m} \left[\frac{\boldsymbol{\theta}^{i}_t - \boldsymbol{\theta}^{w}_t}{R_i + R_s} + \frac{\boldsymbol{\theta}^{e}_t - \boldsymbol{\theta}^{w}_t}{R_m + R_e} + \gamma \mathbf{F}_{\mathbf{H},t} + \frac{R_i}{R_i + R_s} P^{int}_t + \frac{R_e}{R_e + R_m} P^{ext}_t \right]$$

$$\boldsymbol{\theta}^{i}_{t+1} = \boldsymbol{\theta}^{i}_t + \frac{\Delta T}{c_i} \left[\frac{\boldsymbol{\theta}^{w}_t - \boldsymbol{\theta}^{i}_t}{R_i + R_s} + \frac{\boldsymbol{\theta}^{e}_t - \boldsymbol{\theta}^{i}_t}{R_v} + \frac{\boldsymbol{\theta}^{e}_t - \boldsymbol{\theta}^{i}_t}{R_f} + (1-\gamma) \mathbf{F}_{\mathbf{H},t} + \frac{R_s}{R_i + R_s} P^{int}_t \right]$$

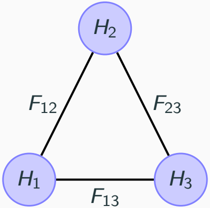
which will be denoted:

$$\boxed{\mathbf{X}_{t+1} = f_t(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1})}$$

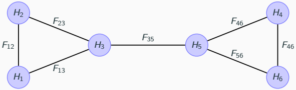
Outline

Viewing the network as a graph

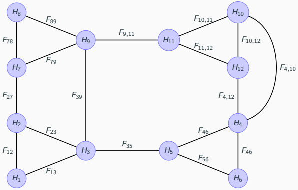
We consider three different configurations



$H1$	House 1	PV + Battery
$H2$	House 2	PV
$H3$	House 3	.

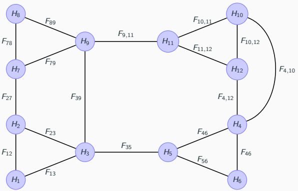


$H1$	House 1	PV + Battery
$H2$	House 2	PV
$H3$	House 3	.
$H4$	House 4	PV + Battery
$H5$	House 5	PV
$H6$	House 6	.



$H1$	House 1	PV + Battery
$H2$	House 2	PV
$H3$	House 3	.
$H4$	House 4	PV + Battery
$H5$	House 5	PV
$H6$	House 6	.
$H7$	House 7	PV + Battery
$H8$	House 8	PV
$H9$	House 9	.
$H10$	House 10	PV + Battery
$H11$	House 11	PV
$H12$	House 12	.

Modeling exchange through the graph



We denote by $\mathbf{Q}$ the flows through the arcs, and $\mathbf{\Delta}$ the balance at the nodes.

The flows must satisfy the Kirchhoff's law:

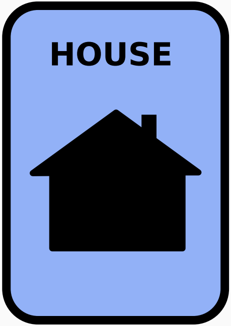
$$A\mathbf{Q} = \mathbf{\Delta}$$

where A is the node-incidence matrix.

We suppose furthermore that losses occur through the arcs ($\eta = 0.96$).

Outline

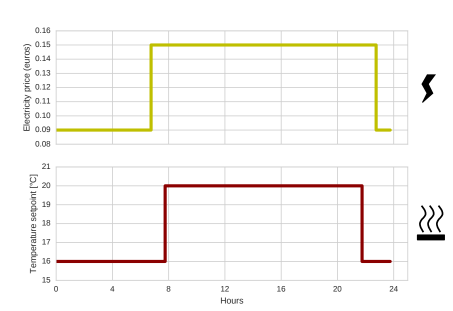
Two commandments to rule them all



Thou shall:

- Satisfy thermal comfort
- Optimize operational costs

Prices and temperature setpoints vary along time



- $T_f = 24\text{h}$, $\Delta T = 15\text{mn}$
- 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

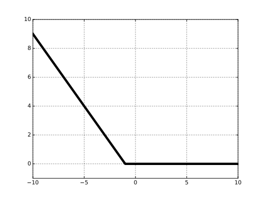
- Cost to import electricity from the network

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

where we define the recourse variable (electricity balance):

$$\underbrace{\mathbf{F}_{NE,t+1}}_{Network} = \underbrace{\mathbf{D}_{t+1}^E}_{Demand} + \underbrace{\mathbf{F}_{\mathbf{B},t}}_{Battery} + \underbrace{\mathbf{F}_{\mathbf{H},t}}_{Heating} + \underbrace{\mathbf{F}_{T,t}}_{Tank} - \underbrace{\mathbf{F}_{pv,t}}_{Solar\ \ panel} + \underbrace{\mathbf{\Delta}_t}_{Exchange}$$

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



κ_{th}

Piecewise linear cost which penalizes temperature if below given setpoint

Instantaneous and final costs for a single house

- The instantaneous convex costs are for the house h

$$L_t^h(\mathbf{X}_t, \mathbf{U}_t, \mathbf{\Delta}_t, \mathbf{W}_{t+1}) = \underbrace{-b_t^E \max\{0, -\mathbf{F}_{NE,t+1}\}}_{buying} + \underbrace{\pi_t^E \max\{0, \mathbf{F}_{NE,t+1}\}}_{selling}$$
$$+ \underbrace{\kappa_{th}(\boldsymbol{\theta}_t^i - \bar{\boldsymbol{\theta}}_t^i)}_{discomfort}$$

- We add a final linear cost

$$K(\mathbf{X}_{T_f}) = -\pi^{\mathbf{H}} \mathbf{H}_{T_f} - \pi^{\mathbf{B}} \mathbf{B}_{T_f}$$

to avoid empty stocks at the final horizon T_f

Writing the stochastic optimization problem

We aim to minimize the costs for all houses

$$\begin{array}{ll} \min\limits_{X,U,Q,\Delta} & \sum_h J^h(X^h, U^h) \\ s.t & AQ = \Delta \end{array}$$

where for each house h:

$$J^h(X^h, U^h, \Delta^h) = \mathbb{E}\left[\sum_{t=0}^{T_f-1} L_t^h(\mathbf{X}_t^h, \mathbf{U}_t^h, \boldsymbol{\Delta}_t^h, \mathbf{W}_{t+1}) + K(\mathbf{X}_{T_f}^h)\right]$$

$$\begin{array}{lll} s.t & \mathbf{X}_{t+1}^h = f_t(\mathbf{X}_t^h, \mathbf{U}_t^h, \mathbf{W}_{t+1}) & \text{Dynamic} \\ & X^\flat \leq \mathbf{X}_t^h \leq X^\sharp & \\ & U^\flat \leq \mathbf{U}_t^h \leq U^\sharp & \\ & X_0^h = X_{ini}^h & \\ & \sigma(\mathbf{U}_t^h) \subset \sigma(\mathbf{W}_1, \dots, \mathbf{W}_t) & \text{Non-anticipativity} \end{array}$$

Resolution methods

Outline

How to solve this stochastic optimal control problem?

We have 96 timesteps (4 x 24) and for each problem

	3 houses	6 houses	12 houses
Stocks	10	20	40
Controls	14	30	68
Uncertainties	8	8	8

How to solve this stochastic optimal control problem?

We have 96 timesteps (4 x 24) and for each problem

	3 houses	6 houses	12 houses
Stocks	10	20	40
Controls	14	30	68
Uncertainties	8	8	8

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 have 96 timesteps (4 x 24) and for each problem

	3 houses	6 houses	12 houses
Stocks	10	20	40
Controls	14	30	68
Uncertainties	8	8	8

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)

MPC vs SDDP: uncertainties modelling

The two algorithms use optimization scenarios to model the uncertainties:

MPC

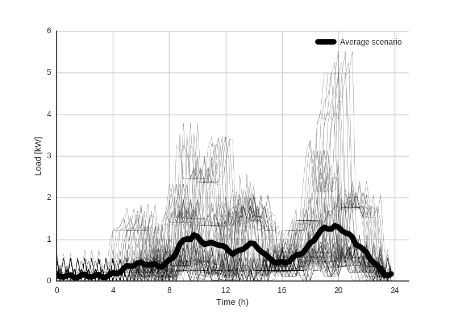


Figure 1: MPC considers the average

...

SDDP

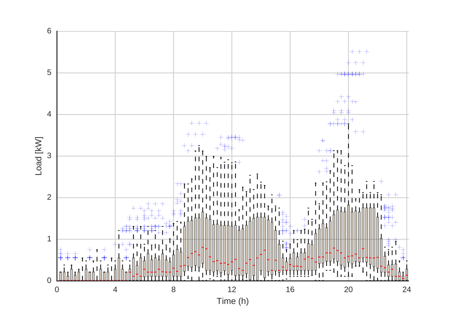


Figure 2: ...and SDDP discrete laws

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) $\left(\overline{W}_{\tau+1}, \dots, \overline{W}_{\tau+H}\right)$
- Solve the **deterministic optimization** problem

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

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

- Get optimal solution $(U^\#_\tau, \dots, U^\#_{\tau+H})$ over horizon $H = 24h$
- Send first control $U^\#_\tau$ to assessor

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**

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

$$\iff \min_{u_\tau} \sum_i \pi_i \Big[L_\tau(X_\tau, u_\tau, W^i_{\tau+1}) + \widetilde{V}_{\tau+1}\Big(f_\tau(X_\tau, u_\tau, W^i_{\tau+1})\Big)\Big]$$

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

A brief recall on Dynamic Programming

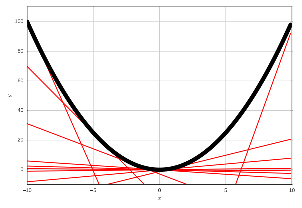
Dynamic Programming

Compute **offline** value functions with the backward equation:

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

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

Stochastic Dual Dynamic Programming



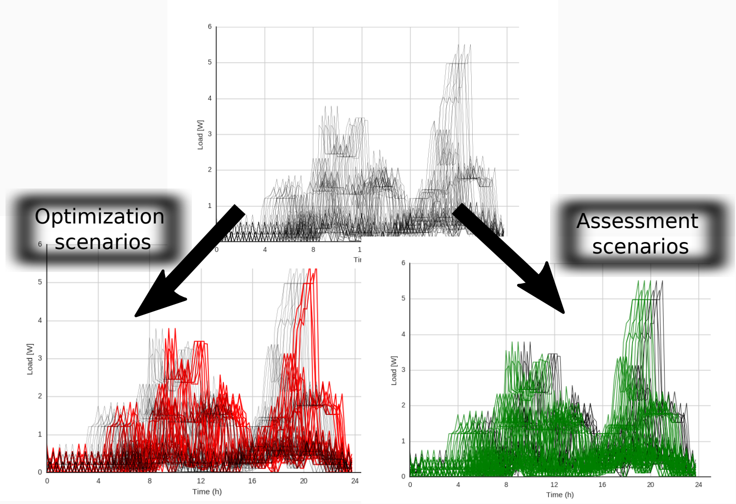
- Convex value functions V_t are approximated as a supremum of a finite set of affine functions
- Affine functions (=cuts) are computed during forward/backward passes, till convergence

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

- SDDP makes an extensive use of LP solver

Outline

Out-of-sample comparison



Numerical resolution

Outline

Our stack is deeply rooted in Julia language

- Modeling Language: `JuMP`

- Open-source SDDP Solver: `StochDynamicProgramming.jl`

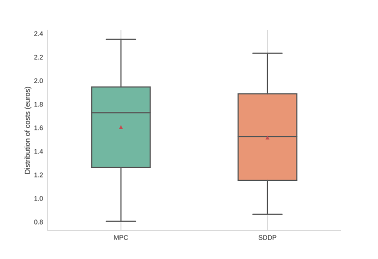
- LP Solver: `Gurobi 7.0`

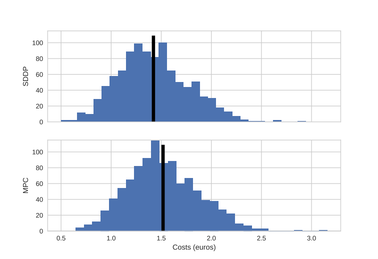
`https://github.com/JuliaOpt/StochDynamicProgramming.jl`

Outline

Comparison of MPC and SDDP

We compare MPC and SDDP over 1000 assessment scenarios



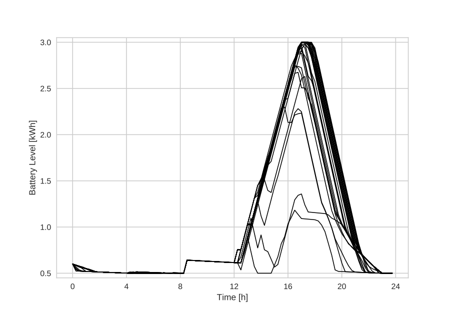


	MPC	SDDP	Diff
	3 houses		
Costs	1.52	1.42	-6.6 %
t_c	0.8	2.8	x3.5
	6 houses		
Costs	3.04	2.85	-6.3 %
t_c	1.7	4.6	x2.7
	12 houses		
Costs	6.08	5.74	-5.6 %
t_c	3.5	8.6	x2.5

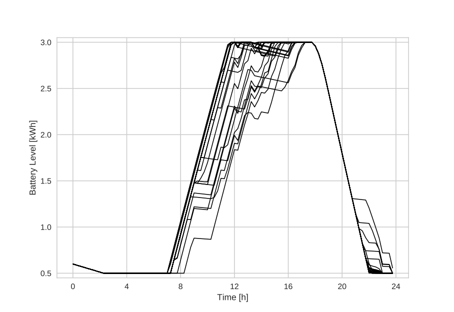
t_c: average time to compute the control online (in ms)

MPC and SDDP use differently the battery

MPC



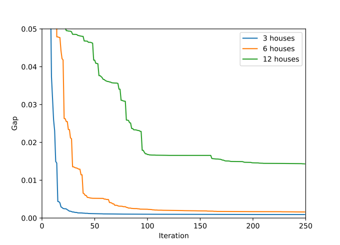
SDDP



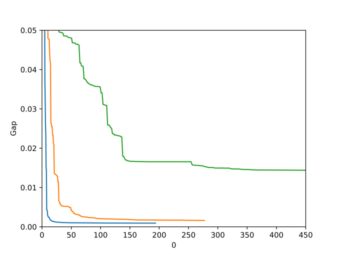
Trajectories of battery for the ‘3 houses’ problem.

Discussing the convergence of SDDP w.r.t. the dimension

We compute the upper-bound afterward, with a great number of scenarios (10000) We define the gap as : $gap = (ub - lb)/ub$.



Gap against number of iterations



Gap against time

We compare the time (in seconds) taken to achieve a particular gap:

gap	3 houses	6 houses	12 houses
2 %	7.0	21.0	137.8
1 %	8.0	28.8	.
0.5 %	8.0	47.2	.
0.1 %	65.1	.	.

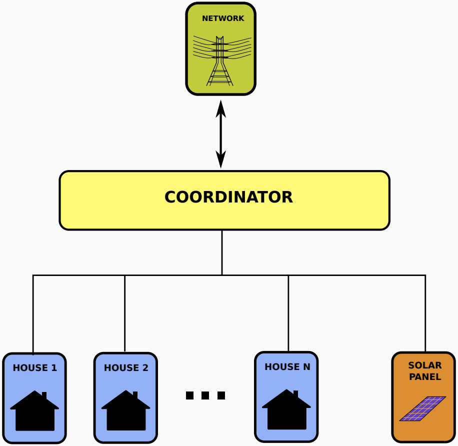
Conclusion

Conclusion

- SDDP beats MPC, however the difference narrows along the number of dimensions (because of the convergence of SDDP)
- Both MPC and SDDP are penalized if dimension becomes too high

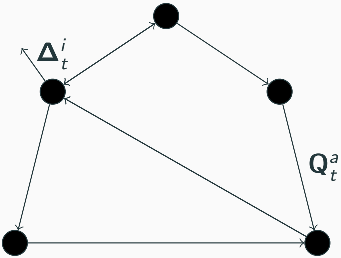
Perspectives

Mix SDDP with spatial decomposition like *Dual Approximate Dynamic Programming* (DADP) to control bigger urban neighbourhood (from 10 to 100 houses)



Modeling exchanges between houses

The grid is represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. At each time $t \in [\![0, T-1]\!]$ we have:



- a flow $\mathbf{Q}_t^a$ through each arc a, inducing a cost $c_t^a(\mathbf{Q}_t^a)$, modeling the exchange between two houses
- a grid flow $\mathbf{\Delta}_t^i$ at each node i, resulting from the balance equation

$$\mathbf{\Delta}_t^i = \sum_{a \in input(i)} \mathbf{Q}_t^a - \sum_{b \in output(i)} \mathbf{Q}_t^b$$

A transport cost decoupled in time

At each time step $t \in [\![0, T-1]\!]$, we define the transport cost as the sum of the cost of the flows $\mathbf{Q}_t^a$ through the arcs a of the grid:

$$J_{,t}[\mathbf{Q}_t] = \mathbb{E}\Big(\sum_{a \in \mathcal{A}} c_t^a(\mathbf{Q}_t^a)\Big) ,$$

where the c_t^a's are easy to compute functions (say quadratic).

Kirchhof's law

The balance equation stating the conservation between $\mathbf{Q}_t$ and $\mathbf{\Delta}_t$ rewrites in the following matrix form:

$$A\mathbf{Q}_t + \mathbf{\Delta}_t = 0 ,$$

where A is the node-arc incidence matrix of the grid.

The overall production transport problem

The *production cost* J_P aggregates the costs at all nodes i:

$$J_P[\mathbf{\Delta}] = \sum_{i \in \mathcal{N}} J_P^i[\mathbf{\Delta}^i] \,,$$

and the *transport cost* J_T aggregates the costs at all time t:

$$J_T[\mathbf{Q}] = \sum_{t=0}^{T-1} J_{T,t}[\mathbf{Q}_t] \,.$$

The compact production-transport problem formulation writes:

$$\min_{\mathbf{Q},\mathbf{\Delta}} \quad J_P[\mathbf{\Delta}] + J_T[\mathbf{Q}]$$
$$\text{s.t.} \;\; A\mathbf{Q} + \mathbf{\Delta} = 0 \,.$$

Introducing decomposition methods

The decomposition/coordination methods we want to deal with are iterative algorithms involving the following ingredients.

- Decompose the global problem in several subproblems of smaller size by processing the constraint $A\mathbf{Q} + \mathbf{\Delta} = 0$,
- Coordinate at each iteration the subproblems using either a price or an allocation.

$$A\mathbf{Q} + \underbrace{\mathbf{\Delta}}_{\textit{allocation}} = 0 \quad \rightsquigarrow \underbrace{\boldsymbol{\lambda}}_{\textit{price}}$$

- Solve the subproblems using Dynamic Programming (when a state is available in the subproblem), taking into account the price or the allocation transmitted by the coordination.

Wandering inside the zoology of decomposition algorithm

Once the problem formulated, it remains to solve it!

- Primal and dual decomposition (via L-BFGS update),
- Operator splitting schemes (ADMM, proximal decomposition, ...),
- Stochastic (accelerated?) gradient descent.

Still a work in progress! ;-)