Artificial intelligence & energy transition: a few stories about energy prosumer communities

Journée scientifique de RISEGrid @ CentraleSupelec

November 30th, 2017

Raphaël Fonteneau @R_Fonteneau

Joint work with D. Ernst, V. François-Lavet, D. Marulli, F. Olivier, D. Taralla






The Smart (Micro) Grids Lab
University of Liège - Belgium

20 researchers developing **algorithmic solutions** for solving research questions in the fields of **microgrids**, **smartgrids**, **globalgrids** & **energy prosumer communities.**

Happy to be back to Supelec :)

Supelec 2007 (Gif 2004-06, Rennes 2006-07)

Ingénierie des Systèmes Automatisés

PhD Thesis in Engineering Science, ULiège, 2011

Contributions to Batch Mode Reinforcement Learning

Inria Lille - Nord Europe - 2012-2013

Analysing the exploration/exploitation dilemma

Back to ULiège - 2013 - …

Conciliating reinforcement learning with the energy transition

A few words about energy prosumer communities

Tentative definition:

« Consumers/Prosumers that organise themselves in order to optimize how they produce and consume energy in order to achieve an objective. »

Many types of energy prosumer communities:

- Physical communities, e.g. people living under the same low-voltage feeder,
- Mobile communities, e.g. people owning EV, able to adapt how and where they charge / discharge their EV,
- Hybridation of these two types of communities.

Outline

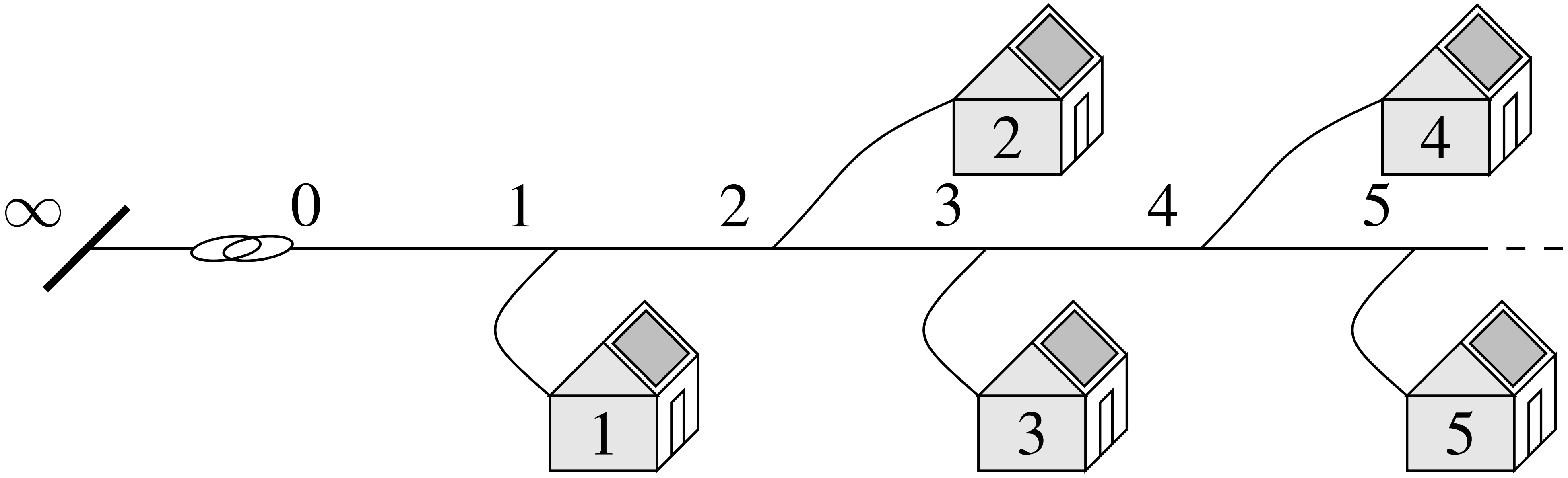
New control challenge within energy prosumer communities

Deep reinforcement learning solutions for energy microgrids

First story

Foreseeing New Control Challenges in Electricity Prosumer Communities

The physical prosumer community



Modeling

N prosumers dynamically interacting with each other over a time horizon *T*:

$$\forall (i,j) \in \{1,\ldots,N\}^2, \forall t \in \{0,\ldots,T-1\},$$
$$\theta_t^{(i \rightarrow j)} = \theta_t^{(j \leftarrow i)}$$

with the convention that $\theta_t^{(i \rightarrow i)} = 0, \theta_t^{(i \leftarrow i)} = 0 \ \forall i, t.$

$$\forall t, i, \ P_{P,t}^{(i)} = L_{P,t}^{(i)} + D_{P,t}^{(i)} + S_{P,t}^{(i)}$$

$$\forall t, i, \ D_{P,t}^{(i)} = \sum_{j=1}^{N} \left(\theta_t^{(i \rightarrow j)} - \theta_t^{(i \leftarrow j)} \right) + \delta D_{P,t}^{(i)}$$

where $\delta D\_P^{(i)}_{P,t}$ is the difference between the power injected into the distribution network and the sum of active power exchanges between the members of the community.

Modeling

Conservation of reactive power at the prosumer's location

$$\forall t, i,\ \ P_{Q,t}^{(i)} = L_{Q,t}^{(i)} + D_{Q,t}^{(i)}$$

Other power-related constraints:

$$\forall t, i\ P_{P,t}^{(i)} \leq P_{P,t}^{(i),\max} \qquad \forall t, i\ \ \left|P_{Q,t}^{(i)}\right| \leq P_{Q}^{(i),\max}\left(P_{P,t}^{(i)}\right)$$

$$\forall t, i,\ \ \left|S_t^{(i)}\right| \leq S^{(i),\max}\left(\lambda_t^{(i)}\right) \qquad \forall t, i\quad 0 \leq \lambda_t^{(i)} \leq \lambda^{(i),\max}$$

Modeling

Network voltage

$$\forall t, i, \qquad \left| D_{P,t}^{(i)} \right| \leq D_P^{(i),\max} \left(P_{P,t}^{(i)}, P_{Q,t}^{(i)}, L_{P,t}^{(i)}, L_{Q,t}^{(i)}, S_t^{(i)}, D_{P,t}^{(j \neq i)}, D_{Q,t}^{(j \neq i)} \right)$$

$$\forall t, i, \qquad \left| D_{Q,t}^{(i)} \right| \leq D_Q^{(i),\max} \left(P_{P,t}^{(i)}, P_{Q,t}^{(i)}, L_{P,t}^{(i)}, L_{Q,t}^{(i)}, S_t^{(i)}, D_{P,t}^{(j \neq i)}, D_{Q,t}^{(j \neq i)} \right)$$

Modeling

Losses « at the root of the community » :

$$\forall t \quad \Lambda_{P,t}^{(c)} = D_{P,t}^{(c)} - \sum_{i=1}^{N} D_{P,t}^{(i)}$$

$$\forall t \quad \Lambda_{Q,t}^{(c)} = D_{Q,t}^{(c)} - \sum_{i=1}^{N} D_{Q,t}^{(i)}$$

where $D^{(c)}_{P,t}$ (resp. $D^{(c)}_{Q,t}$) is the active (resp. reactive) power measured at the root of the community.

Modeling

Costs and revenues

$$c_t^{(i)} = \Delta\Bigg(\max\left(-\delta D_t^{(i)}, 0\right) Pr_t^{(D\to i)} + \sum_{j=1}^{N} \max\left(\theta_t^{(i\leftarrow j)}, 0\right) Pr_t^{(j\to i)} \Bigg)$$

$$r_t^{(i)} = \Delta\Bigg(\max\left(\delta D_t^{(i)}, 0\right) Pr_t^{(i\to D)} + \sum_{j=1}^{N} \max\left(\theta_t^{(j\leftarrow i)}, 0\right) Pr_t^{(j\to i)} \Bigg)$$

Modeling

State and prices vector

$$\Xi_t = \begin{pmatrix} P_{P,t}^{(1)} & P_{Q,t}^{(1)} \\ P_{P,t}^{(1),\max} & P_{Q,t}^{(1),\max} \\ \vdots & \vdots \\ P_{P,t}^{(N)} & P_{Q,t}^{(N)} \\ P_{P,t}^{(N),\max} & P_{Q,t}^{(N),\max} \\ S_t^{(1)} & \lambda_t^{(1)} \\ \vdots & \vdots \\ S_t^{(N)} & \lambda_t^{(N)} \\ L_{P,t}^{(1)} & L_{Q,t}^{(1)} \\ \vdots & \vdots \\ L_{P,t}^{(N)} & L_{Q,t}^{(N)} \\ D_{P,t}^{(1)} & D_{Q,t}^{(1)} \\ \vdots & \vdots \\ D_{P,t}^{(N)} & D_{Q,t}^{(N)} \end{pmatrix}, \quad \Phi_t = \begin{pmatrix} Pr_t^{(D\to 1)} \\ Pr_t^{(1\to D)} \\ \vdots \\ Pr_t^{(D\to N)} \\ Pr_t^{(N\to D)} \\ Pr_t^{(1\to 2)} \\ Pr_t^{(2\to 1)} \\ \vdots \\ Pr_t^{(1\to N)} \\ Pr_t^{(N\to 1)} \\ \vdots \\ Pr_t^{(N-1\to N)} \\ Pr_t^{(N\to N-1)} \end{pmatrix}$$

$$\Theta_t^{\rightarrow} = \left(\theta_t^{(i\rightarrow j)}\right)_{i,j}, \qquad \Theta_t^{\leftarrow} = \left(\theta_t^{(i\leftarrow j)}\right)_{i,j}$$

$$\Xi_{t+1} = F(\Xi_t, \Phi_t, \Theta_t^{\rightarrow}, \Theta_t^{\leftarrow} \ldots, \Xi_0, \Phi_0, \Theta_0^{\rightarrow}, \Theta_0^{\leftarrow}, \omega_t)$$

$$\omega_t \sim P_t(\cdot)$$

Controlling the prosumers community

Need for an optimisation criterion

Local production:

$$\max_{\substack{P_{P,t}^{(i)}, P_{Q,t}^{(i)}, L_{P,t}^{(i)}, L_{Q,t}^{(i)}, S_t^{(i)}, \Theta_t^{\rightarrow}, \Theta_t^{\leftarrow} \\ t \in \{0,\dots,T-1\} \\ i \in \{1,\dots,N\}}} \mathbb{E}\left[\sum_{t=0}^{T-1} \sum_{i=1}^{N} P_{P,t}^{(i)}\right]$$

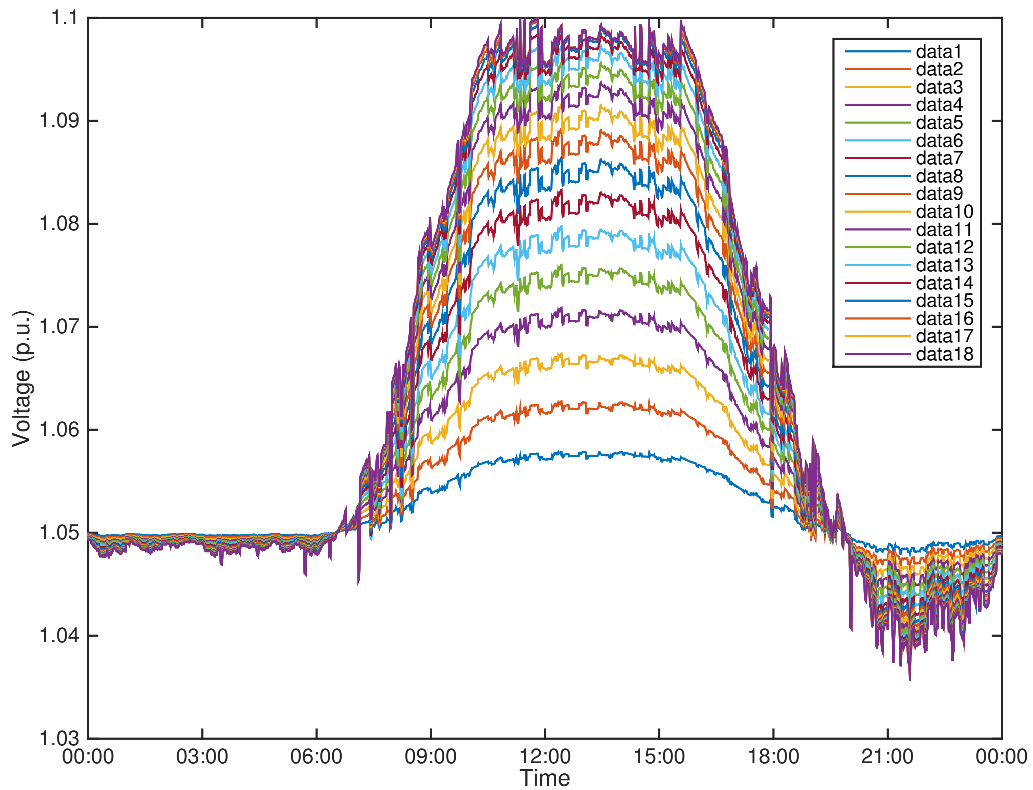
Taking losses into account:

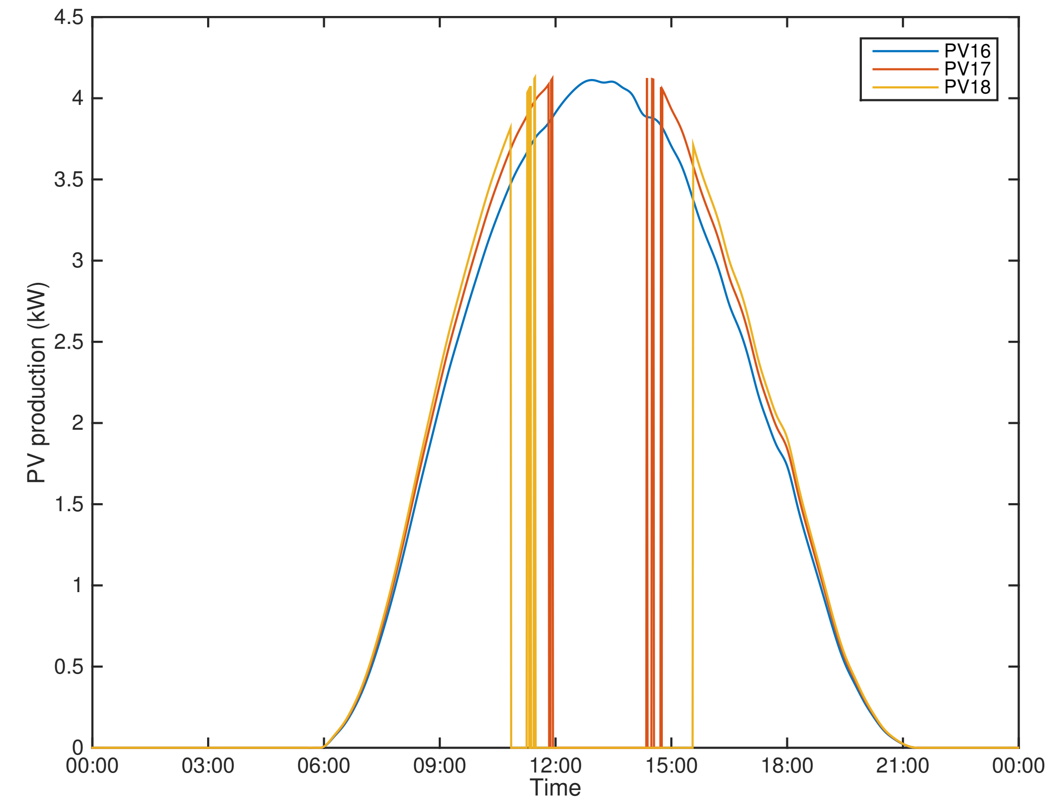
$$\max_{\substack{P_{P,t}^{(i)}, P_{Q,t}^{(i)}, L_{P,t}^{(i)}, L_{Q,t}^{(i)}, S_t^{(i)}, \\ \Theta_t^{\rightarrow}, \Theta_t^{\leftarrow} \\ t \in \{0,\dots,T-1\} \\ i \in \{1,\dots,N\}}} \mathbb{E}\left[\sum_{t=0}^{T-1} \sum_{i=1}^{N} P_{P,t}^{(i)} - \Lambda_{P,t}^{(c)}\right]$$

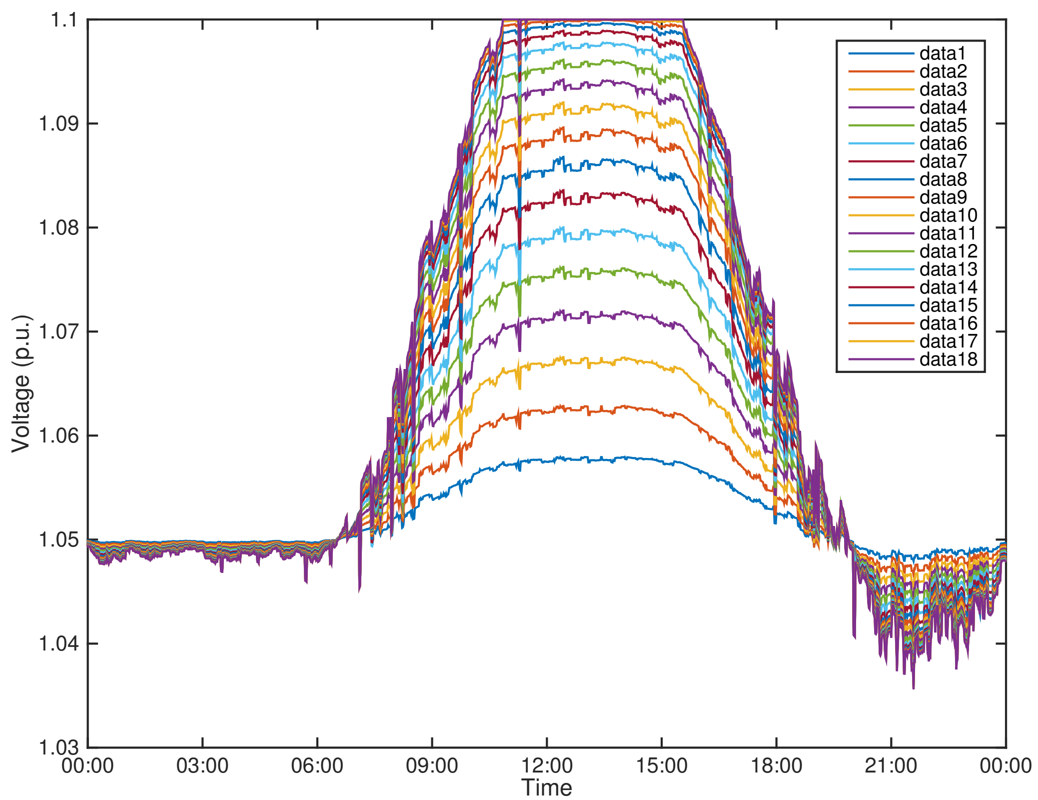
Revenues minus costs:

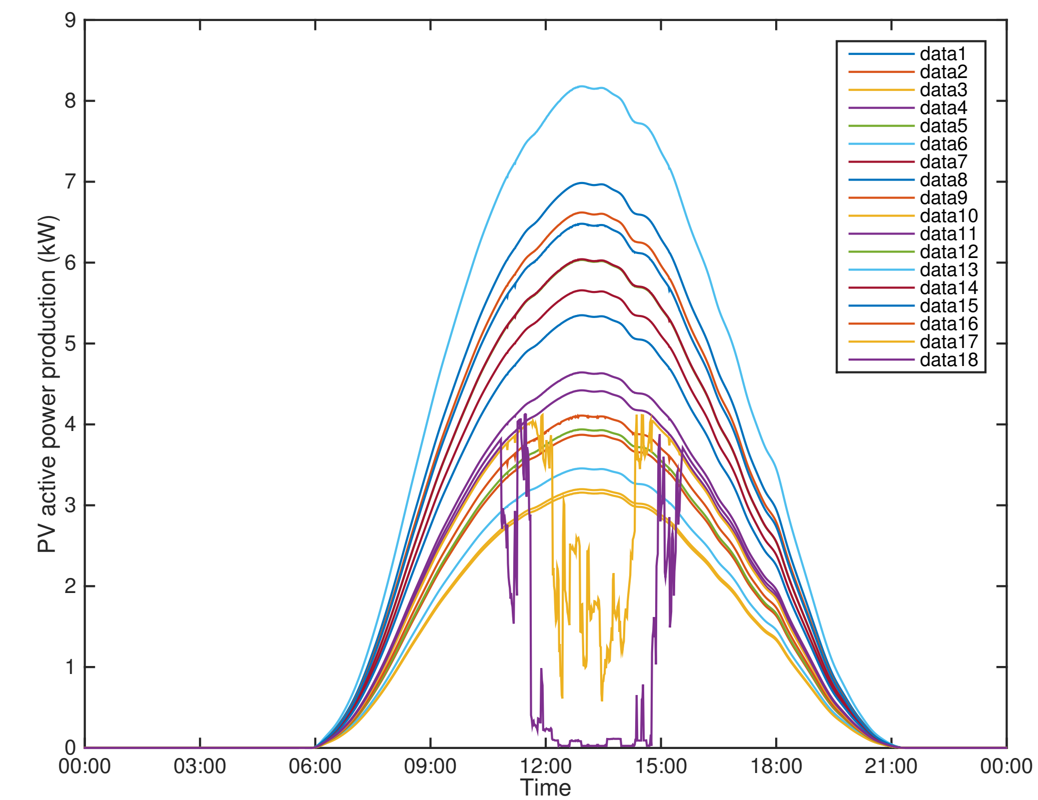
$$\max_{\substack{P_{P,t}^{(i)}, P_{Q,t}^{(i)}, L_{P,t}^{(i)}, L_{Q,t}^{(i)}, S_t^{(i)}, \Theta_t^{\rightarrow}, \Theta_t^{\leftarrow} \\ t \in \{0,\dots,T-1\} \\ i \in \{1,\dots,N\}}} \mathbb{E}\left[\sum_{t=0}^{T-1} \sum_{i=1}^{N} r_t^{(i)} - c_t^{(i)}\right]$$

Centralised control without storage (P)
Optimising PV production









Centralised control with storage

A centralised strategy

- We use the forward backward sweep optimal power flow strategy proposed in [Fortenbacher et al.]

- We obtain both a sizing and centralised planning strategy, for a given load and solar irradiance scenario

Optimal Sizing and Placement of Distributed Storage in Low Voltage Networks. Philipp Fortenbacher Martin Zellner Göran Andersson. IEEE Power Systems Computation Conference (PSCC), 2016.

Why going decentralised?

Technical challenges for building centralised strategies

1. Information gathering
2. Need for a centralised controller for processing information
3. Concretising computational results into applied actions

Learning a decentralised strategy

We propose a data-driven, « learning approach »:

1. Built a set of centralised solutions

2. Generate learning (input, output) samples, where the input is made from local indicators, and the output is a decision that should be applied locally

3. Learn a strategy from the samples

    - Imitative learning

Building a set of data

First, generate scenarios: a solar irradiance scenarios, a set of load scenarios (for each prosumer).

Then, solve the pairs {solar irradiance, load profiles} (using, for instance, a forward backward sweep power flow approach)

$$\left[\Xi_0^*, \ldots, \Xi_{T-1}^*\right]$$

From this time series of data, one can extract a series of local data, i.e. relative to one single prosumer (i) :

$$\left[\Xi_0^{(i),*}, \ldots, \Xi_{T-1}^{(i),*}\right]$$

$$\Xi_t^{(i),*} = \begin{pmatrix} P_{P,t}^{(i),*} & P_{Q,t}^{(i),*} \\ P_{P,t}^{(i),\max} & P_{Q,t}^{(i),\max} \\ S_t^{(i),*} & \lambda_t^{(i),*} \\ L_{P,t}^{(i)} & L_{Q,t}^{(i)} \\ D_{P,t}^{(i),*} & D_{Q,t}^{(i),*} \end{pmatrix}$$

Learning from data - Imitative learning

We propose to use a machine learning approach

- Machine learning is about extracting pattern from data
- From the sample of data
  we learn a mapping state -> action

$$L = \left(s^{(i)}, a^{(i)}\right)_{i=1}^{N}$$

Here, we adopt a slightly indirect approach by learning 4 different regressors:

- Active power
- Reactive power
- Charging battery
- Discharging the battery

1st regressor: learning active power production

Data set: $$\mathcal{L}^P = \left\{ \left(in_P^{i,t}, out_P^{i,t} \right) \right\}_{i=1,t=0}^{i=N,t=T-1}$$

$$in_P^{i,t} =$$

$$\left(i,\ |\overline{\mathbf{v}}_t^{(i)}|,\ arg(\overline{\mathbf{v}}_t^{(i),*}),\ \phi_t,\ \lambda_t^{(i),*},\ L_{P,t}^{(i)},\ P_{P,t}^{(i),max} \right)$$

$$out_P^{i,t} = P_{P,t}^{(i),*}$$

- i : id number of the bus
- $|\overline{\mathbf{v}}_t^{(i)}|$: magnitude of the voltage at bus i at time step t
- $arg(\overline{\mathbf{v}}_t^{(i)})$: phase of the voltage at bus i at time step t
- ϕ_t : electricity price at time step t, considered as being unique in the whole feeder
- $\lambda_t^{(i)}$: level of charge of the storage of bus i at time step t
- $L_{P,t}^{(i)}$: load consumption at bus i at time step t
- $P_{P,t}^{(i),max}$: maximal production potential at bus i at time step t

2nd regressor: learning reactive power production

Data set: $$\mathcal{L}^Q = \left\{ \left(in_Q^{i,t}, out_Q^{i,t} \right) \right\}_{i=1,t=0}^{i=N,t=T-1}$$

$$in_Q^{i,t} = in_P^{i,t}$$

$$out_Q^{i,t} = P_{Q,t}^{(i),*}$$

3rd regressor: learning how to charge the battery

Data set:

$$\mathcal{L}^C = \left\{ \left(in_C^{i,t}, out_C^{i,t} \right) \right\}_{i=1,t=0}^{i=N,t=T-1}$$

$$\begin{aligned} in_C^{i,t} &= in_P^{i,t} \\ out_C^{i,t} &= \max\left(S_t^{(i),*}, 0 \right) \end{aligned}$$

4th regressor: learning how to draw power from the battery

Data set:

$$\mathcal{L}^D = \left\{ \left(in_D^{i,t}, out_D^{i,t} \right) \right\}_{i=1,t=0}^{i=N,t=T-1}$$

$$\begin{aligned} in_D^{i,t} &= in_P^{i,t} \\ out_D^{i,t} &= \max\left(-S_t^{(i),*}, 0 \right) \end{aligned}$$

Then, post processing & evaluating solutions

Post-processing solutions

-> Ensure physical constraints are satisfied

- For the active and reactive power production levels, ensure that the production levels are compatible with production bounds, for each prosumer i
- For the power injected into / drawn from the battery, ensure that both maximal charging/discharging powers and of the level of charge evolution are feasible

Generating other scenarios to try the learned strategy

-> Evaluate the performance of learned policies in other environments

Test case

The number of buses is 15

The number of prosumers is 14

The number of branches is 14

Δt is 1h

The time horizon T is 8760

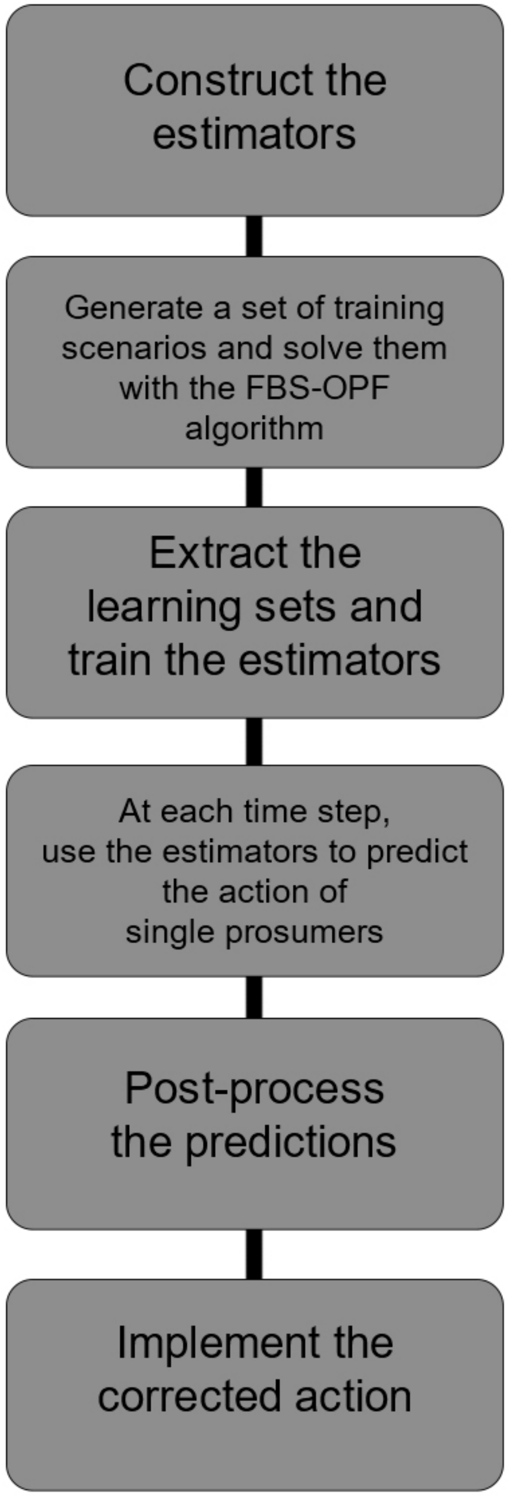
The line resistance $R_{d1} = R_{d2} = ... = R_{dL}$ is 0.025 Ω

The line reactance $X_{d1} = X_{d2} = ... = X_{dL}$ is 0.005 Ω

The nominal voltage of the network is 400 V

The maximum admissible voltage v^{max} is 1.10 *pu*

The minimum admissible voltage v^{min} is 0.90 *pu*



Test case: prosumers characteristics

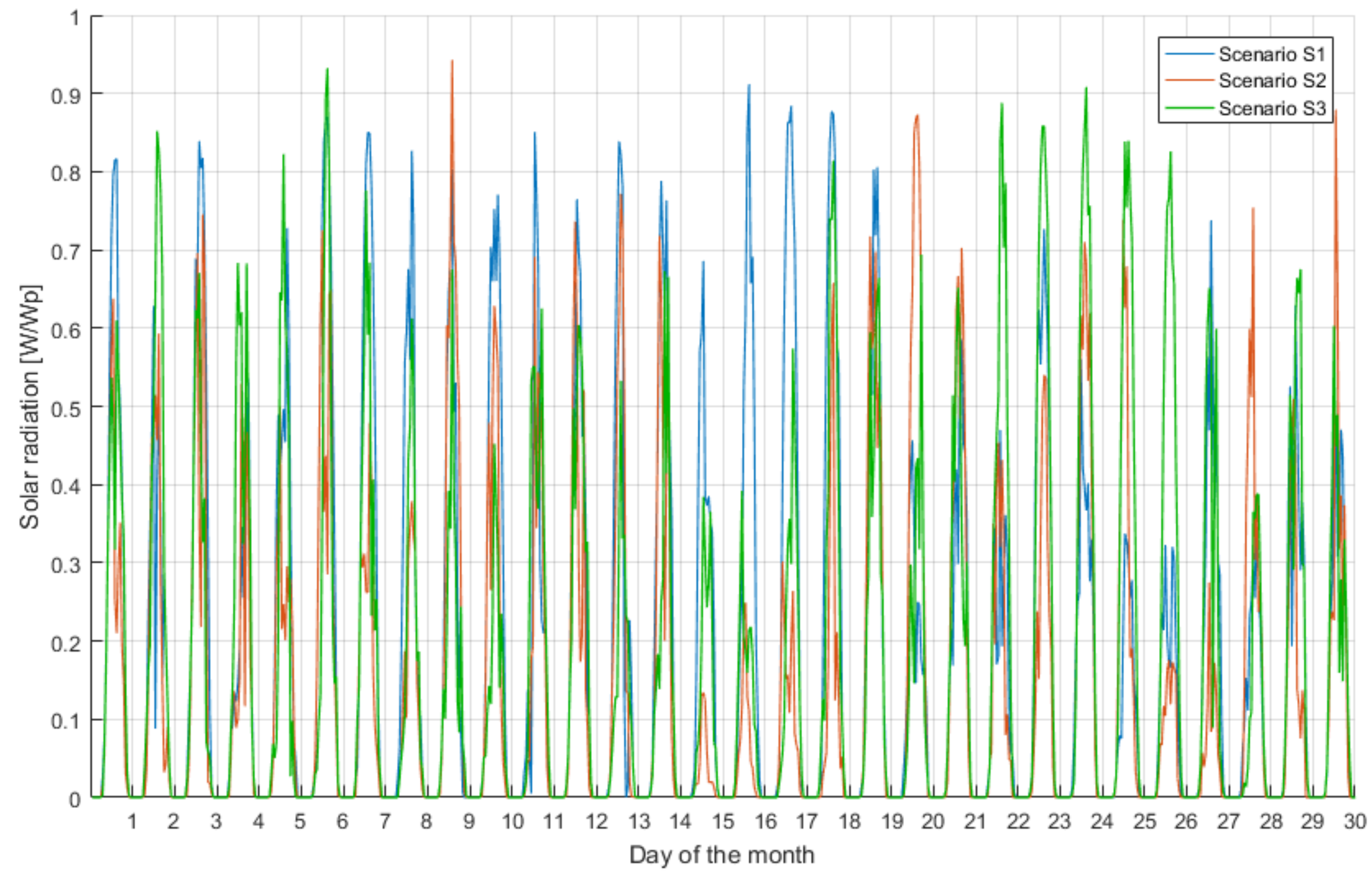
Id	**Number of occupants**	**PV installed capacity** kW_p	**Storage installed capacity** kWh
1	1	2	2
2	1	2	2
3	2	3	2
4	2	3	2
5	2	3	2
6	3	3.5	5
7	3	3.5	5
8	3	3.5	5
9	4	5	6
10	4	5	6
11	4	5	6
12	4	5	6
13	5	7	8
14	5	7	8

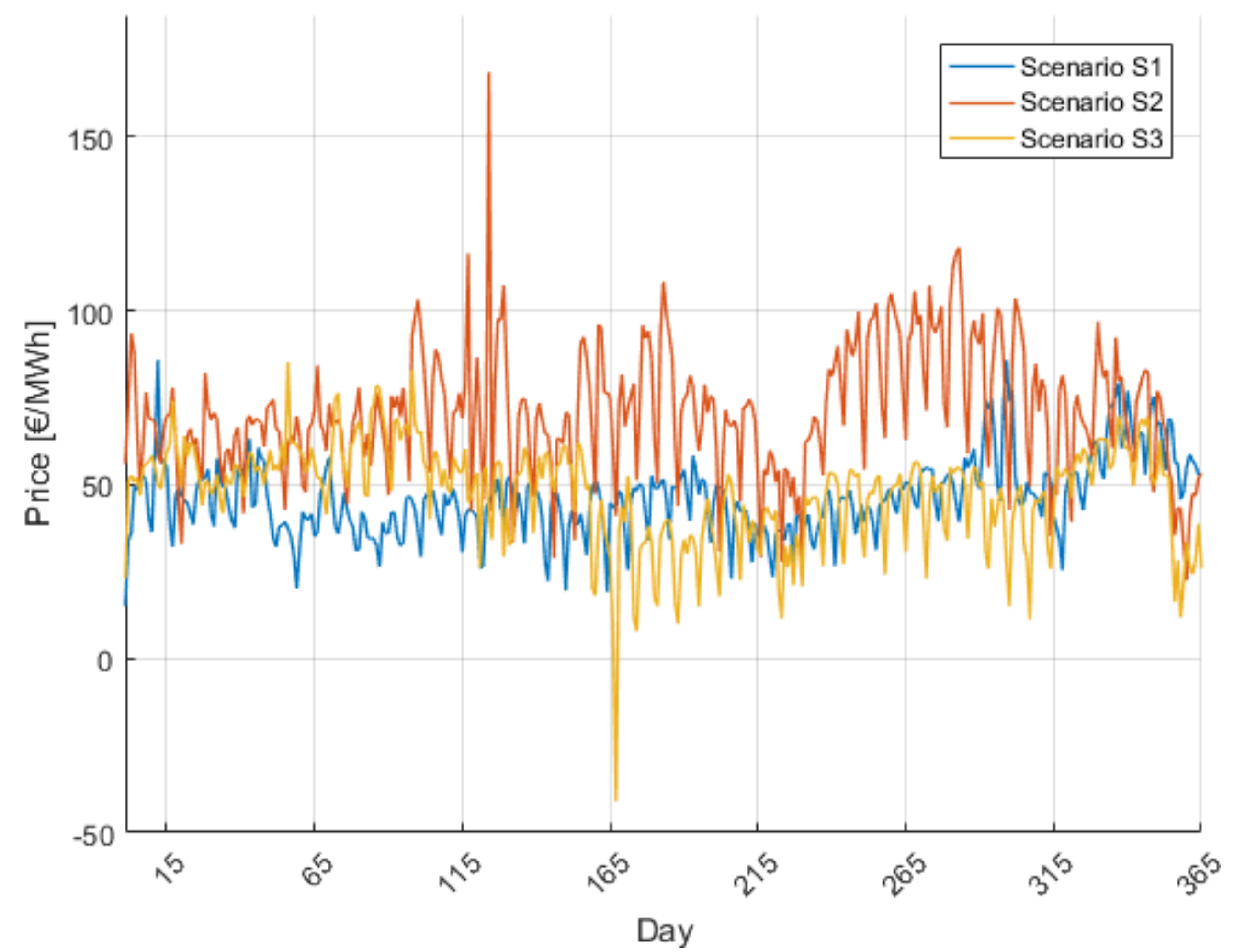
Test scenarios

Load profiles are generated using the model provided in:

Richardson, I., Thomson, M., Infield, D., & Clifford, C. (2010). Domestic electricity use: A high-resolution energy demand model. *Energy and buildings*, *42*(10), 1878-1887.

3 scenarios solar production + electricity prices



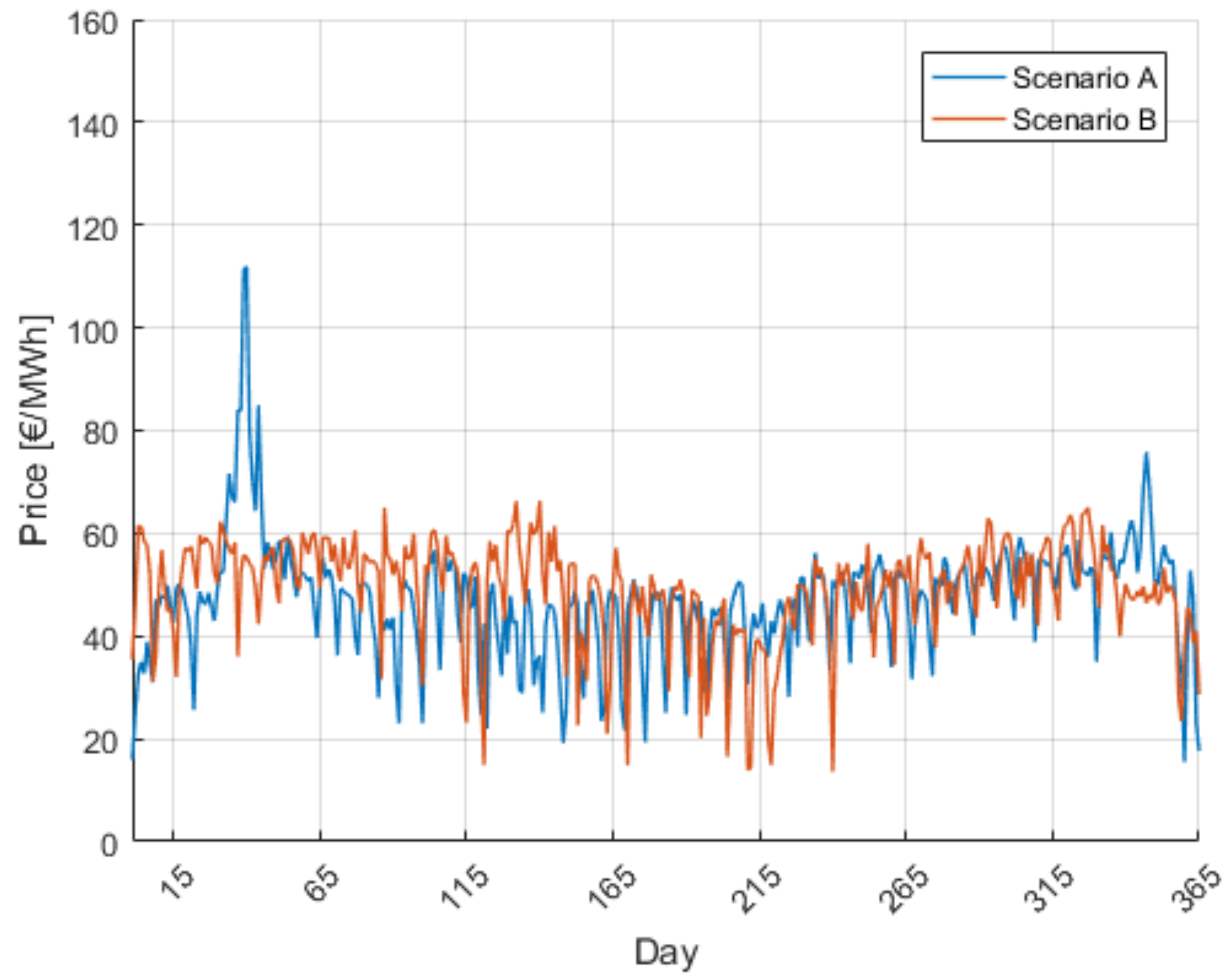


Learning scenarios

We generate two additional price scenarios, S4 and S5.

The FBS-OPF algorithm is run on these two scenarios.

The resulting outputs of the FBS-OPF are used to generate learning sets for the regressors.



Results

Overall costs (objective function)

Overall costs			
Scenario	S1	S2	S3
FBS-OPF algorithm	1105.54 €	2121.16 €	1837.80 €
SL algorithm	2711.44 €	7832.43 €	5123.09 €
RT algorithm	5143.32 €	6501.94 €	5807.77 €

Energy outlook:

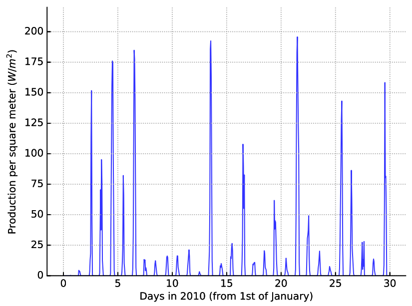
Curtailments over the year			
Scenario	S1	S2	S3
FBS-OPF algorithm	7.01%	11.20%	9.69%
SL algorithm	11.13%	32.78%	14.80%
RT algorithm	11.91%	13.46%	15.12%

Second story

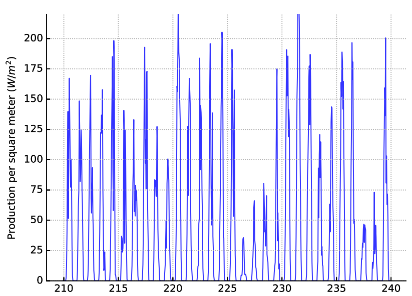
Deep reinforcement learning solutions for energy microgrids management

Operating storage devices in microgrids

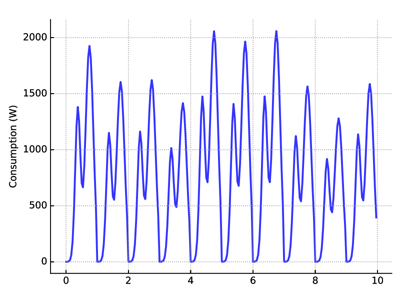
The context: imagine a microgrid (MG) featuring photovoltaic (PV) panels, with both short and long terme storage devices.

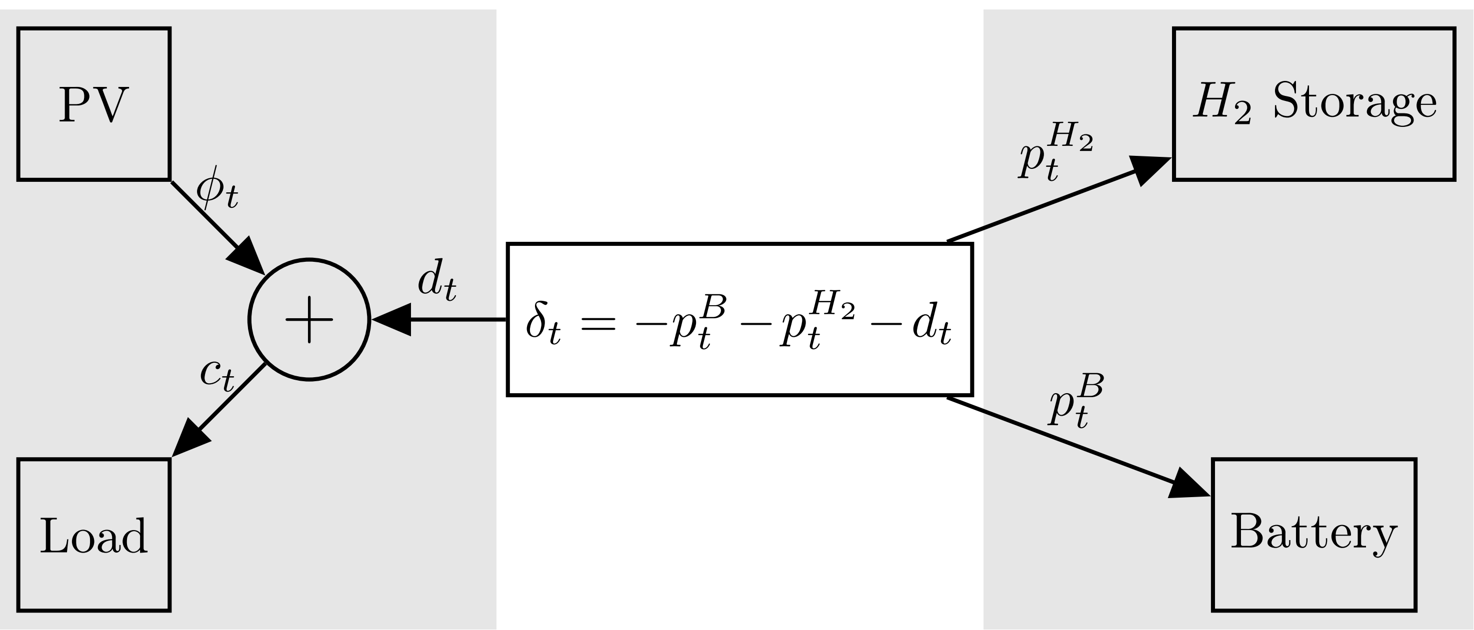


(b) Example of production in winter



(c) Example of production in summer





The problem: how to optimally active the storage devices so that to minimise the operating costs of the MG?

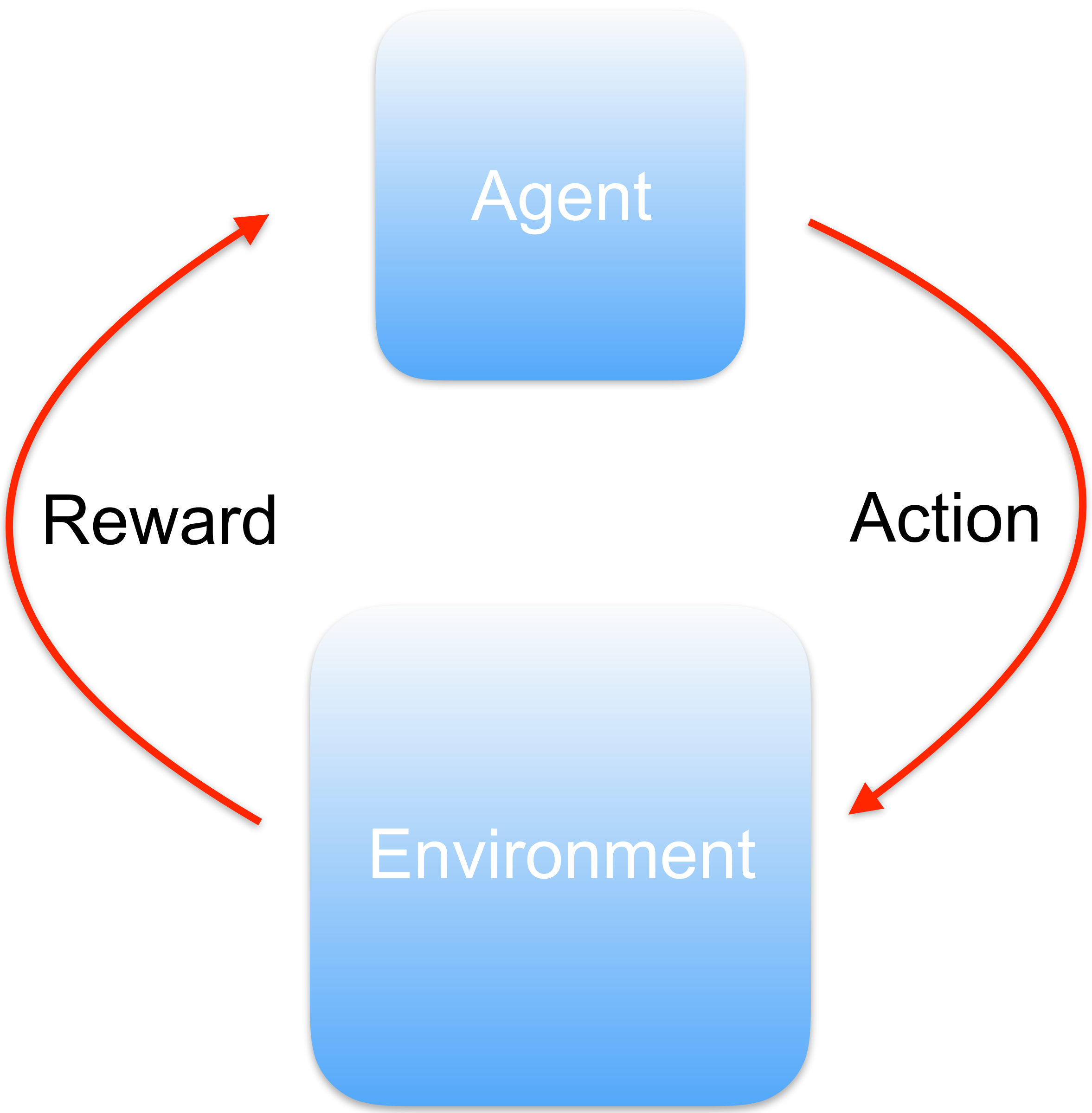
(Deep) Reinforcement Learning (RL)

Ingredients

1) An agent evolving within an environment

2) A reward function, assessing the immediate quality of decision

3) A capacity of interacting with the environment



(Deep) Reinforcement Learning (RL)
More formally…

Markov Decision Process $M = (\mathcal{S}, \mathcal{A}, T, R)$

$$\mathcal{S} = \left\{s^{(1)}, \ldots, s^{(n_{\mathcal{S}})}\right\} \quad \mathcal{A} = \left\{a^{(1)}, \ldots, a^{(n_{\mathcal{A}})}\right\}$$

$$T(s_t, a_t, s_{t+1}) = P(s_{t+1} | s_t, a_t) \qquad r_t = R(s_t, a_t, s_{t+1})$$

Policy: $\pi : \mathcal{S} \rightarrow \mathcal{A}$

$$\forall s \in \mathcal{S}, \quad J^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t), s_{t+1}) \Big| s_0 = s\right]$$

$$\gamma \in [0, 1)$$

(Deep) Reinforcement Learning (RL)
More formally…

Searching for optimality $\forall s \in \mathcal{S}, \quad J^{\pi^*}(s) \geq J^{\pi}(s)$

Solving (or approximating) the Bellman equation:

$$\forall s \in \mathcal{S},$$
$$J^*(s) = \max_{a \in \mathcal{A}} \quad \sum_{s' \in \mathcal{S}} T(s,a,s')\left(R(s,a,s') + \gamma J^*(s')\right)$$

Theoretically, one may just to behave optimally with respect to the optimal state-action value function:

$$Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$$

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A},$$
$$Q^*(s,a) = \sum_{s' \in \mathcal{S}} T(s,a,s')\left[R(s,a,s') + \gamma J^*(s')\right]$$

In practice: partial observability, too many states / dimensions…

Formalising a RL problem

- State : $s_t = \big[[c_{t-h^c}, \dots, c_{t-1}], [\phi_{t-h^p}, \dots, \phi_{t-1}], s_t^{MG}\big]$

  $s_t = \big[[c_{t-h^c}, \dots, c_{t-1}], [\phi_{t-h^p}, \dots, \phi_{t-1}], s_t^{MG}, \zeta_s\big]$

  $s_t = \big[[c_{t-h^c}, \dots, c_{t-1}], [\phi_{t-h^p}, \dots, \phi_{t-1}], s_t^{MG}, \zeta_s, \rho_{24}, \rho_{48}\big]$

- Action: $a_t = [a_t^{H_2}, a_t^B] \in \mathcal{A}_t$ $\quad \forall t \in \mathcal{T}$: $\mathcal{A}_t = \Big([-\zeta^B s_t^B, \frac{x^B - s_t^B}{\eta^B}]\Big) \times \Big([-\zeta^{H_2} s_t^{H_2}, \infty[\cap [-x^{H_2}, x^{H_2}]\Big)$

- Battery dynamics: $s_{t+1}^B = s_t^B + \eta_t^B a_t^B$ if $a_t^B \geq 0$ and $s_{t+1}^B = s_t^B + \frac{a_t^B}{\zeta_t^B}$ otherwise.

- Hydrogen dynamics: $s_{t+1}^{H_2} = s_t^{H_2} + \eta_t^{H_2} a_t^{H_2}$ if $a_t^{H_2} \geq 0$ and $s_{t+1}^{H_2} = s_t^{H_2} + \frac{a_t^{H_2}}{\zeta_t^{H_2}}$ otherwise.

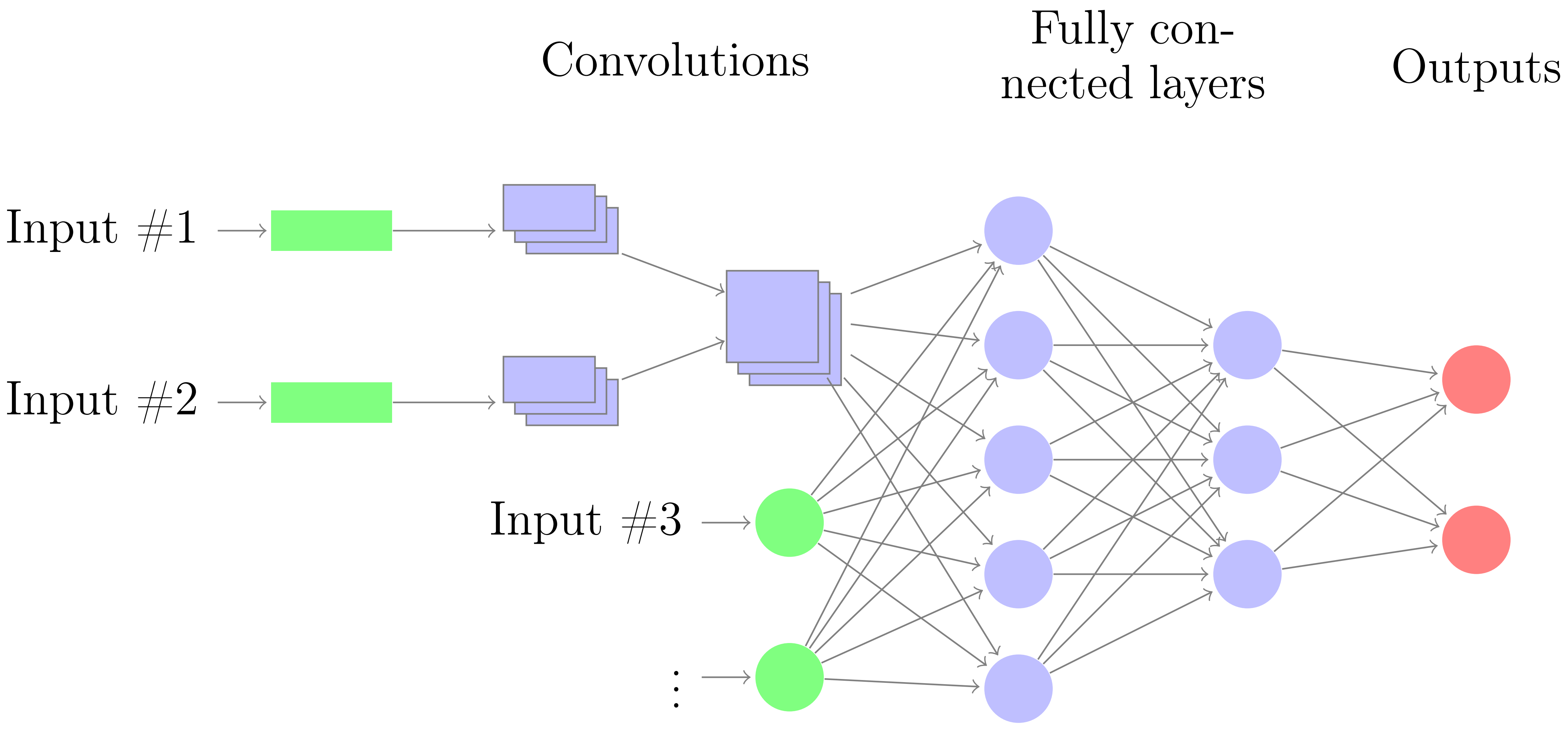
- Reward function: $r_t = r(a_t, d_t) = r^{H_2}(a_t, d_t) + r^-(a_t, d_t)$

$$r^-(a_t, d_t) = k\delta_t \text{ when } \delta_t < 0$$

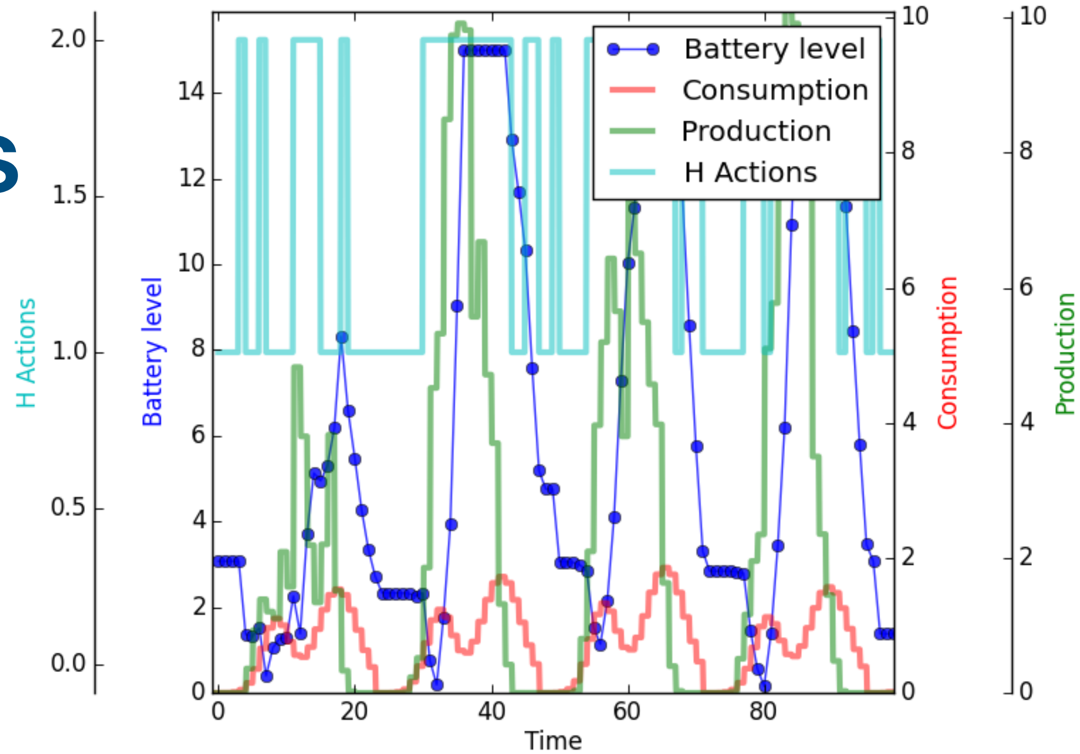
$$r^{H_2}(a_t, d_t) = k^{H_2} a_t^{H2}$$

Using a deep neural network to approximate the value function

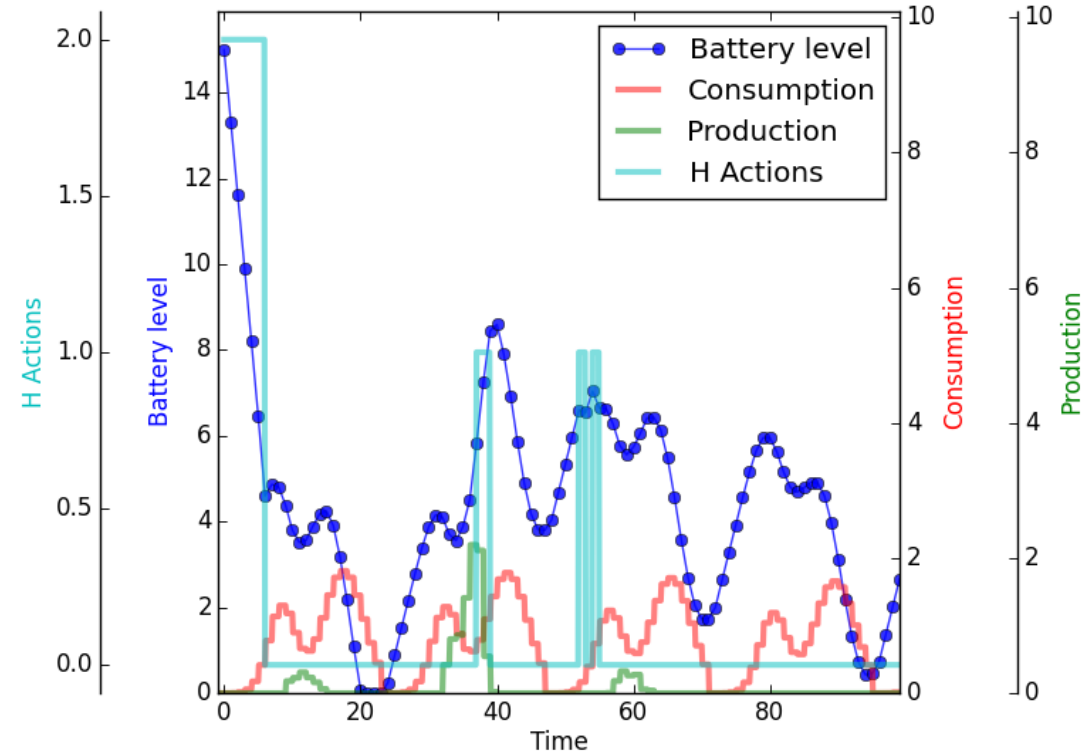
Approximating a state-action value function using a deep neural network.



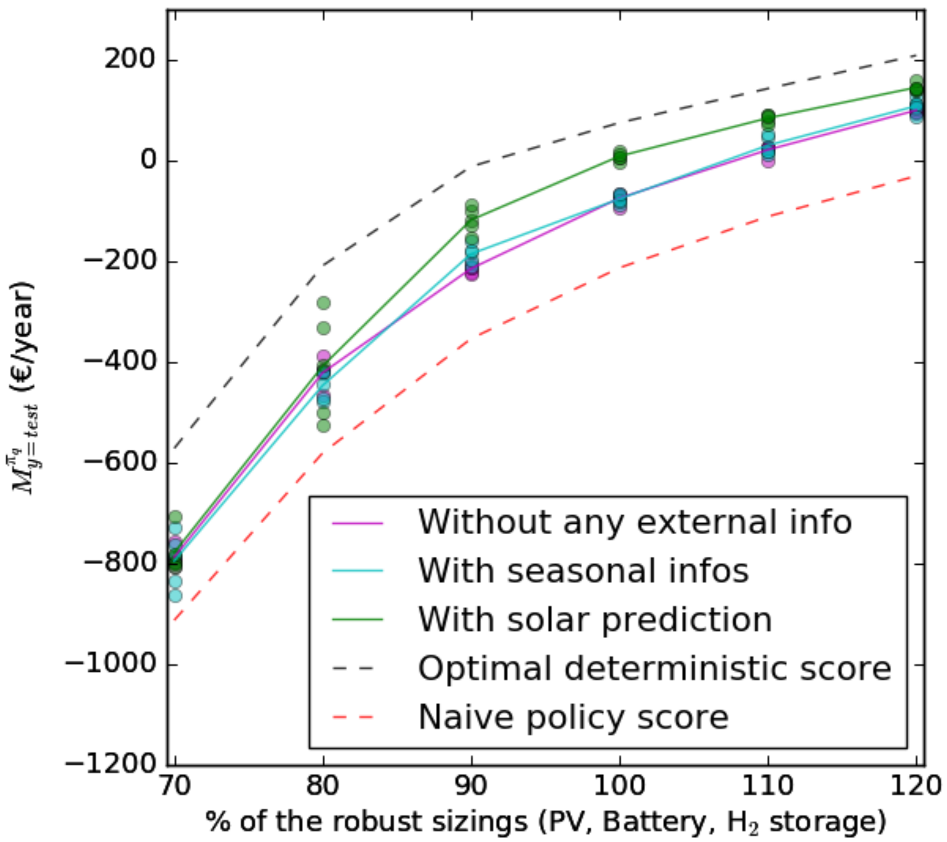
Results



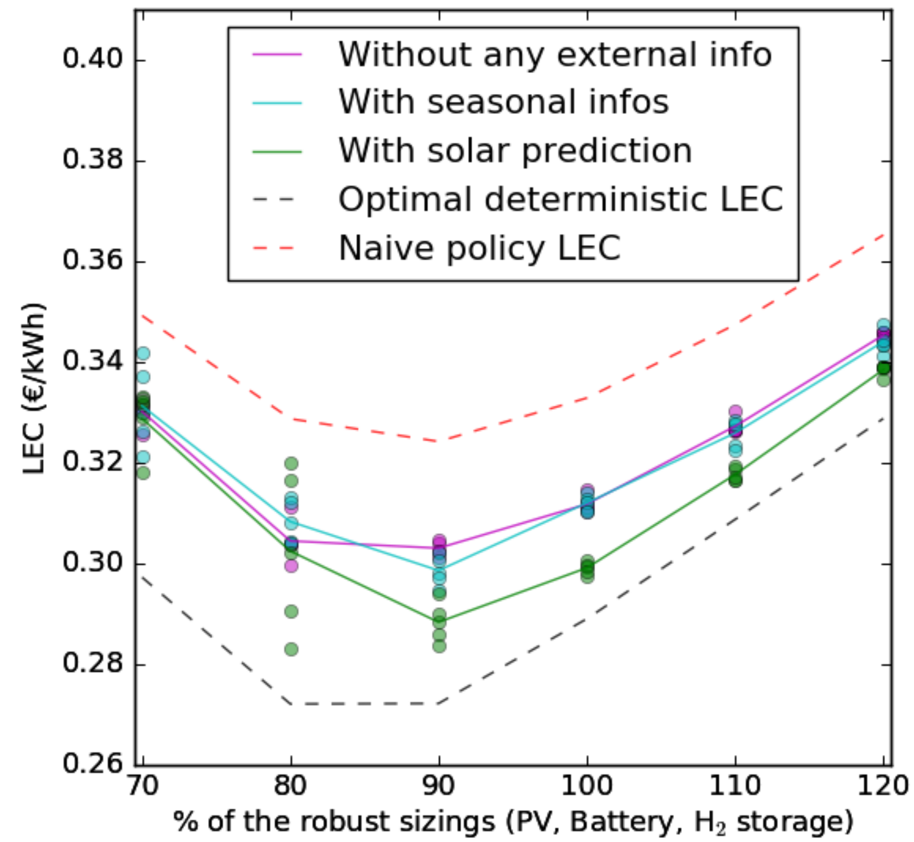
(a) Typical policy during summer



(b) Typical policy during winter



(a) Operational revenue



(b) LEC

V. François-Lavet open source project: the DeeR framework

DeeR (Deep Reinforcement) is a python library to train an agent how to behave in a given environment so as to maximise a cumulative sum of rewards:

https://github.com/VinF/deer

As a conclusion…

(Ongoing) next steps

Many, many problems to (re)think regarding energy prosumer communities

We are currently working on the integration of (distributed) reinforcement learning approaches for agents to cooperate within a community

Also, we are investigating how to take into account 3 phase unbalanced load problems…

References

[1] *Towards the minimization of the levelized energy costs of microgrids using both long-term and short-term storage devices*
V. François-Lavet, Q. Gemine, D. Ernst, R. Fonteneau
Smart Grid: Networking, Data Management, and Business Models, 295-319, 2016

[2] *Deep reinforcement learning solutions for energy microgrids management*
V. François-Lavet, D. Taralla, D. Ernst, R. Fonteneau
European Workshop on Reinforcement Learning (EWRL), 2016

[3] *An app-based algorithmic approach for harvesting local and renewable energy using electric vehicles*
A. Dubois, A. Wehenkel, R. Fonteneau, F. Olivier, D. Ernst
International Conference on Agents and Artificial Intelligence (ICAART), 2017

[4] *Automatic phase identification of smart meter measurement data*
F. Olivier, D. Ernst, R. Fonteneau
Congrès International des Réseaux Electriques de Distribution (CIRED), 2017

[5] *Foreseeing New Control Challenges in Electricity Prosumer Communities*
F. Olivier, D. Marulli, D. Ernst, R. Fonteneau
10th Bulk Power Systems Dynamics and Control Symposium (IREP), 2017

[6] *Reinforcement Learning for Electric Power System Decision and Control: Past Considerations and Perspectives*
M. Glavic, R. Fonteneau, D. Ernst
The 20th World Congress of the International Federation of Automatic Control (IFAC), 2017