Smart online planning in microgrids

Samy Aittahar

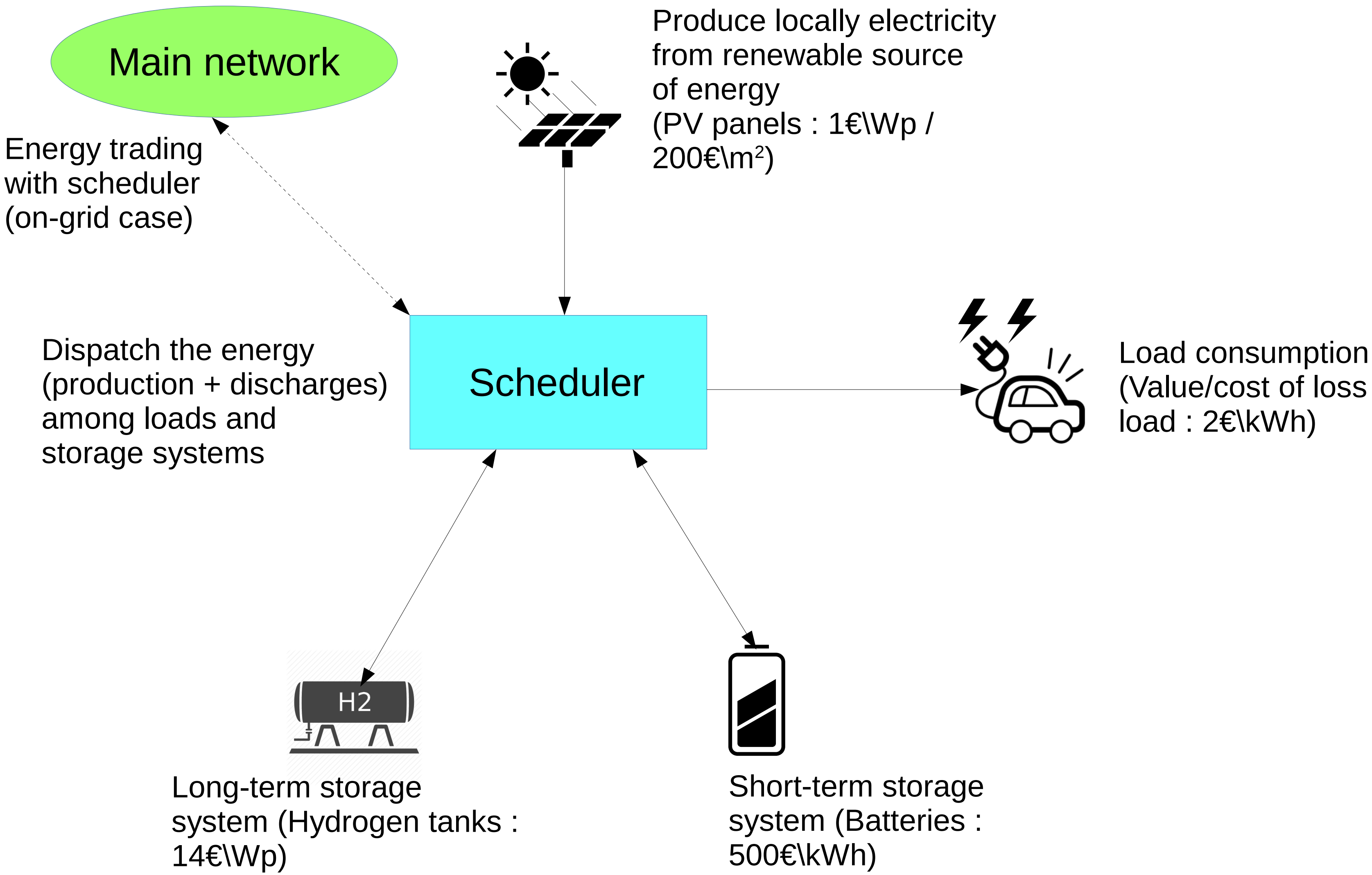
saittahar@student.ulg.ac.be
Phd Student
Supervisor : Pr Damien Ernst
Montefiore Institute (SYSTMOD)
University of Liège (ULg)

Presentation for DARE (ECML-PKDD) 2015 of following paper :
Imitative learning for online planning in microgrids [1]

ECML PKDD SEP 07 11 TEM BER 2015

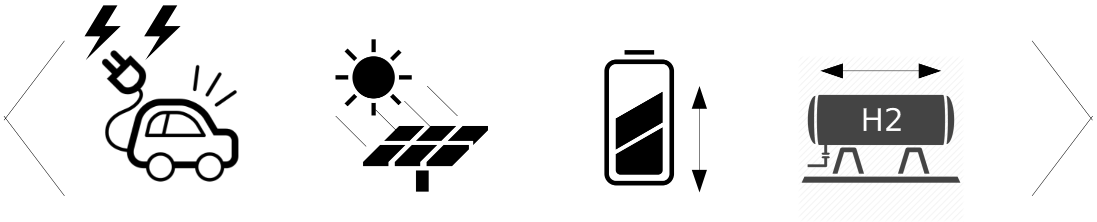
Université de Liège

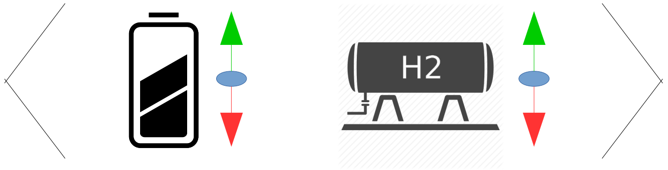
Microgrid


Main network
Energy trading
with scheduler
(on-grid case)
Produce locally electricity
from renewable source
of energy
(PV panels : 1€\Wp /
200€\m²)
Dispatch the energy
(production + discharges)
among loads and
storage systems
Scheduler
Load consumption
(Value/cost of loss
load : 2€\kWh)
H2
Long-term storage
system (Hydrogen tanks :
14€\Wp)
Short-term storage
system (Batteries :
500€\kWh)

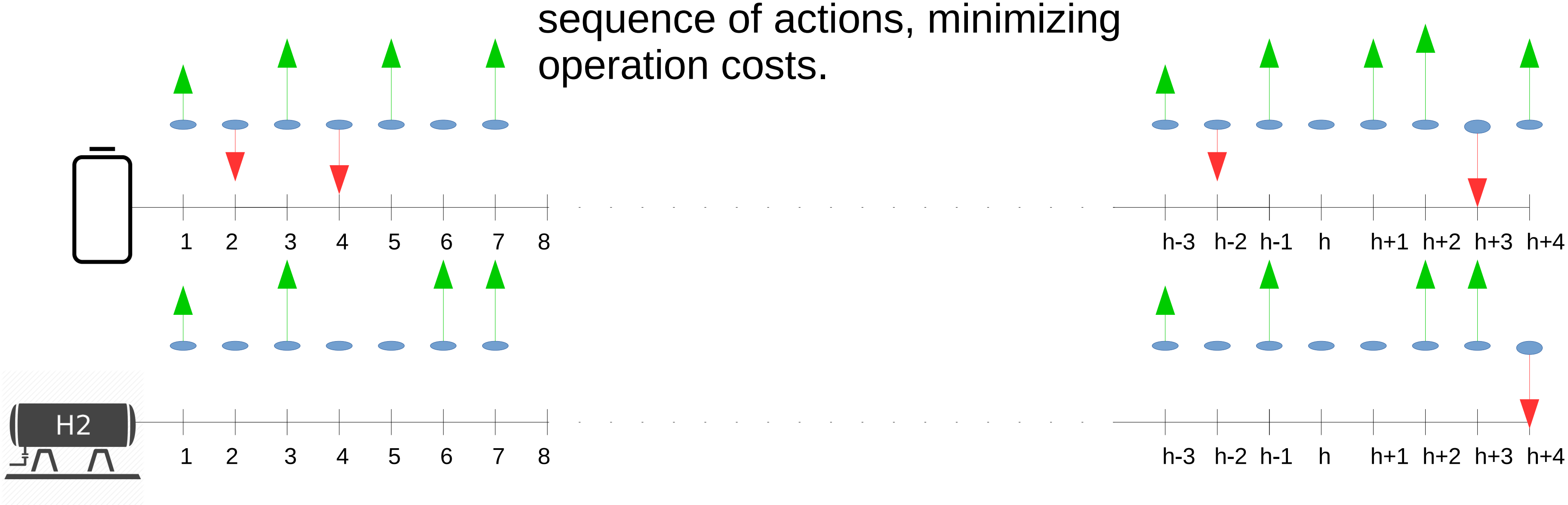
Planning a microgrid

Given prior informations and possible actions at each time step,

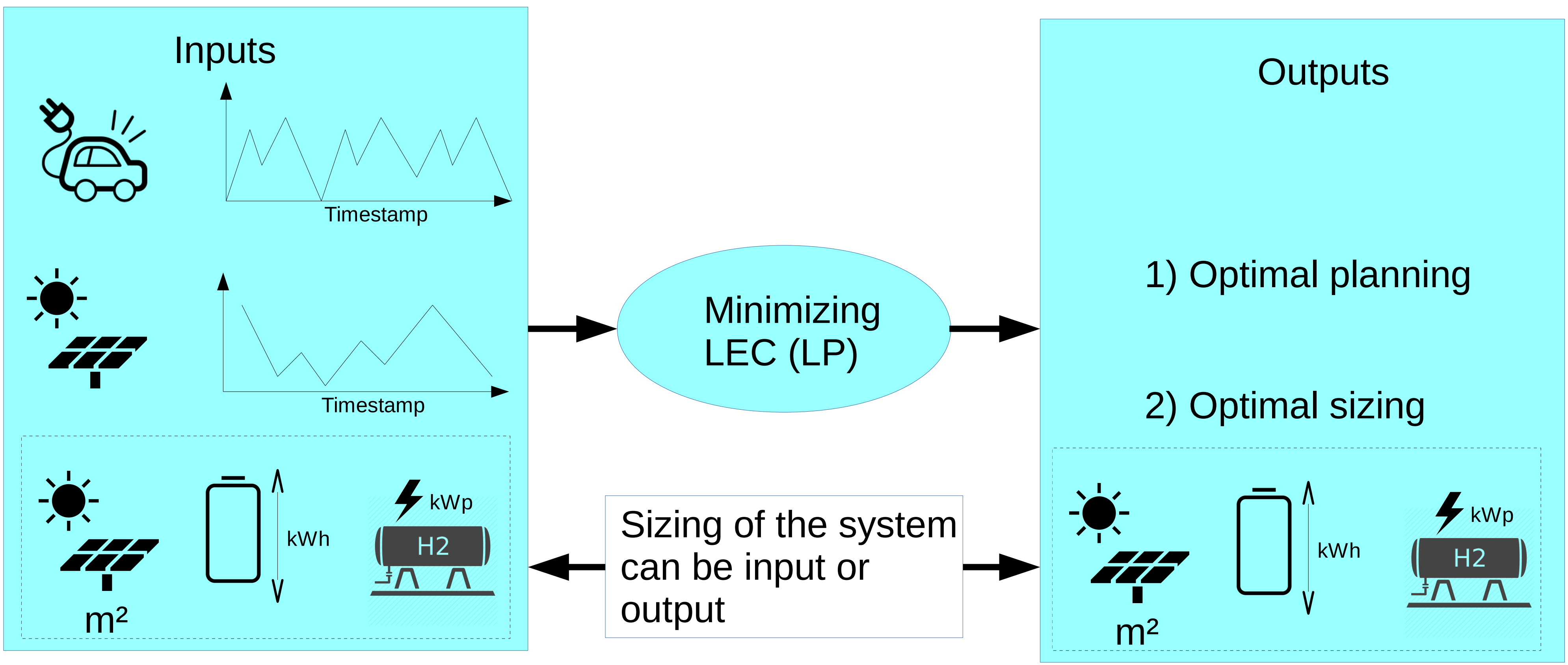




We aim to determine a smart sequence of actions, minimizing operation costs.



Easy planning when future is known



$$LEC = \frac{\sum_{t=1}^{T} \frac{-\sum_{\psi\in\Psi} k_t^{\psi} F_t^{\psi}}{(1+r)^y} + I_0}{\sum_{y=1}^{n} \frac{\epsilon_y}{(1+r)^y}}$$

Where

$k_t^{\psi} \longrightarrow$ Value of loss load

$F_t^{\psi} \longrightarrow$ Amount of loss load

$I_0 \longrightarrow$ Initial investment

Why is it easy ?

$$\text{Min.} \ \frac{\sum_{t=1}^{T} \frac{-\sum_{\psi \in \Psi} k_t^{\psi} F_t^{\psi}}{(1+r)^y} + I_0}{\sum_{y=1}^{n} \frac{\epsilon_y}{(1+r)^y}}, y = t/(365 \times 24)$$

$$\text{S.t.}, \forall t \in \{0 \dots T-1\} :$$

$$s_t^{\sigma} = s_{t-1}^{\sigma} + a_{t-1}^{-,\sigma} + a_{t-1}^{+,\sigma}, \forall \sigma \in \Sigma \ ,$$

Storage systems dynamics

$$s_t^{\sigma_c} \leqslant x^{\sigma_c}, \forall \sigma_c \in \Sigma_C \ ,$$

$$a_t^{+,\sigma_p} \leqslant x^{\sigma_p}, \forall \sigma_p \in \Sigma_P \ ,$$

$$a_t^{-,\sigma_p} \leqslant x^{\sigma_p}, \forall \sigma_p \in \Sigma_P \ ,$$

Storage systems limits
(content or/and power limit)

$$\sum_{\psi \in \Psi} F_t^{\psi} \leqslant -d_t - \sum_{\sigma \in \Sigma} \eta^{\sigma} (a_t^{-,\sigma} + a_t^{+,\sigma}) \ ,$$

$$\sum_{\psi \in \Psi} F_t^{\psi} \leqslant 0 \ ,$$

$$-F_t^{\psi} \leqslant c_t^{\psi} \ .$$

Value of loss load

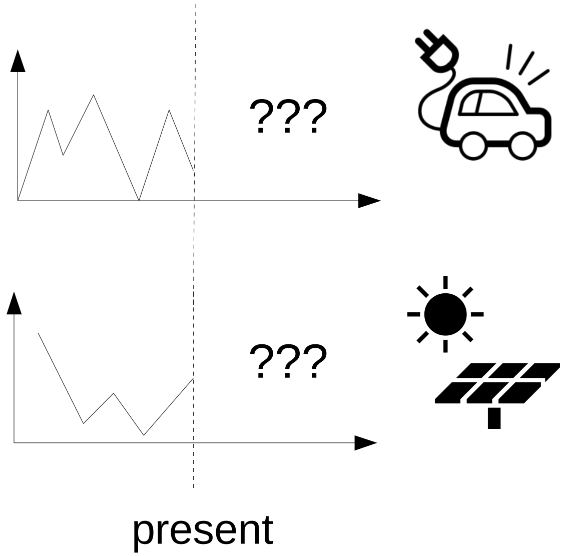
c_t^{ψ} Consumption of load ψ

ϕ_t^{g} Production of generator g

$$d_t = c_t - \sum_{g \in G} \phi_t^{g}, \forall 0 \leqslant t \leqslant T-1$$

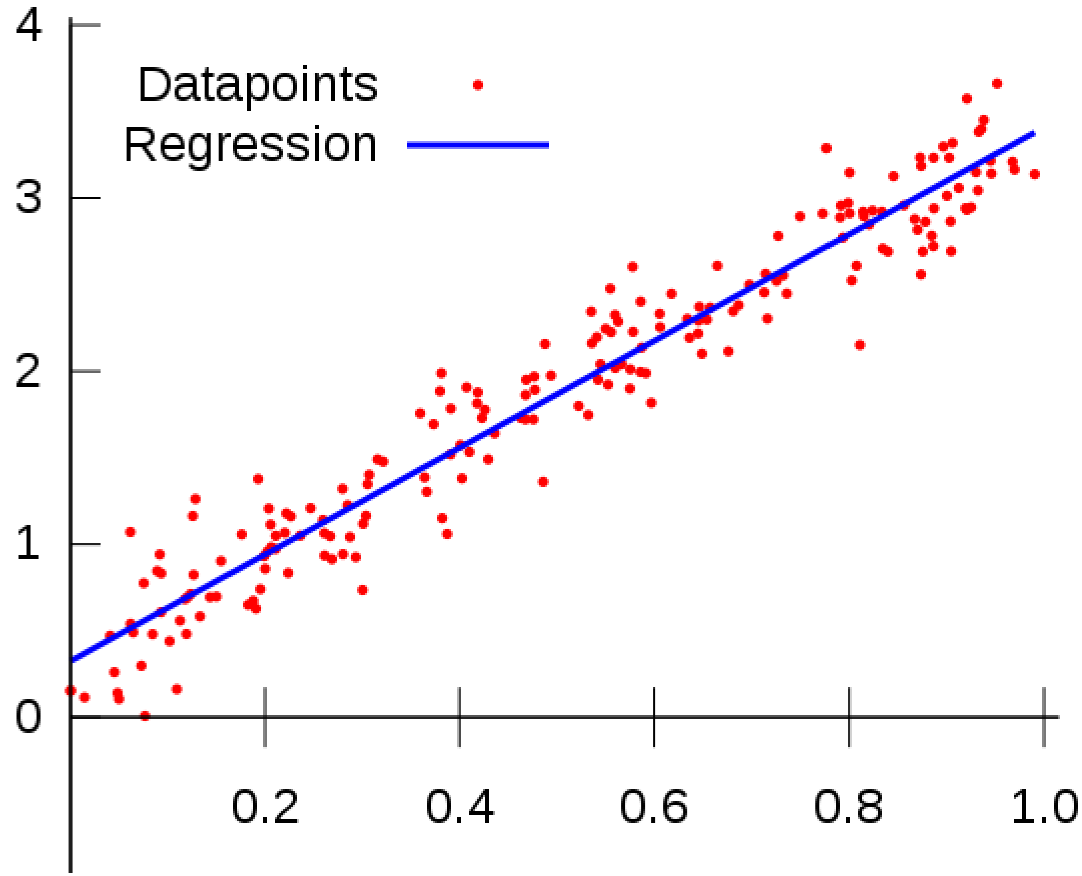
Implementation of the microgrid dynamics as a linear program (inspired from V.François-Lavet et al., business case study results on microgrids [2]). Such a program can be solved efficiently using Simplex algorithm.

But, in real life...



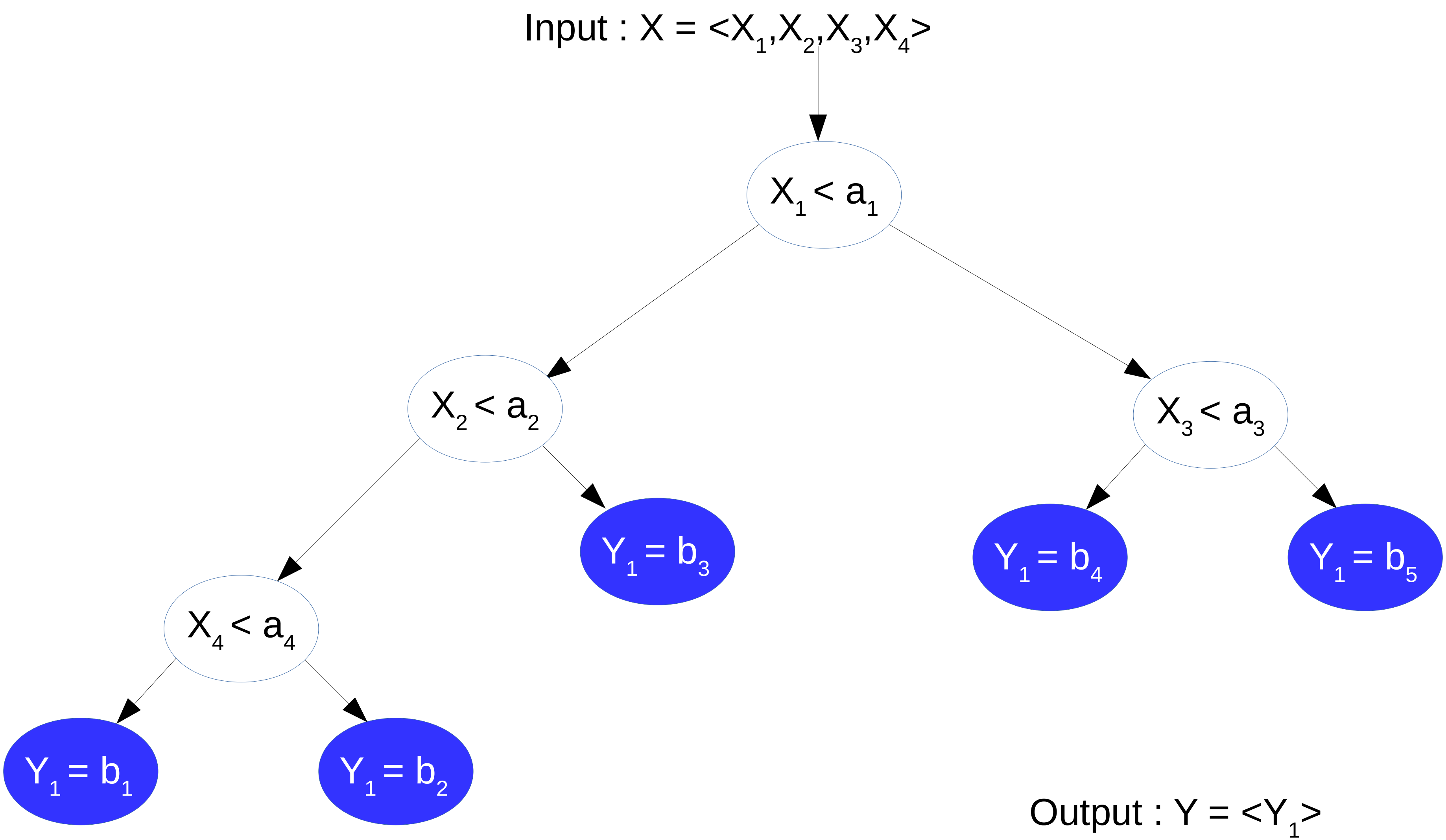
- Designing a smart scheduler (i.e. performing a near-optimal planning) is complex ;
- Informations about future consumption and production should be provided by the history of consumption and production ;
- And remember, generating a large optimal database is easy ;
- How to use such data ?

Introduction to Machine Learning



- Machine learning allows to build more complex systems from larger input space
- Input space has great influence on the quality of the built system.

Example of learning structure : Decision Tree



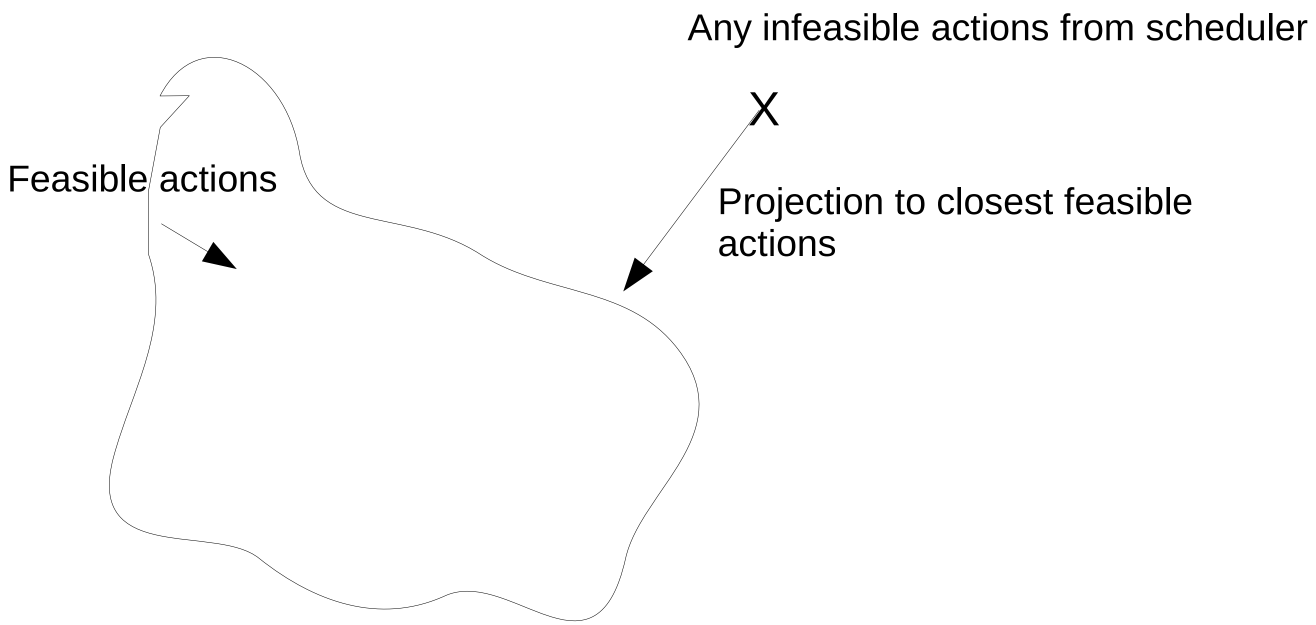
Algorithm : Extremely Randomized Trees developed by Geurts et al. [3]

Building the input space

- Any efficient scheduler needs informative data from the microgrid ;
- Need to choose smartly the attributes for building an efficient learning structure from :
  - Informations from a subwindow of production and consumption history ;
  - Current datetime (e.g. season). ;
- The learning set will contains data from sequences of actions, described by such attributes ;
- The scheduler will be built from such a learning set as a function able to choose actions, given informations extracted from the microgrid ;
- Once such a function is built, the scheduler will make use of it to perform planning in microgrids ;
- This is **imitative learning**.

Ensure output consistency

- The scheduler can possibly apply actions independently on each storage system. May lead to inconsistencies.
- Scheme below show how to guarantee compliance of actions with constraints.



How to Ensure output consistency

$$\text{Min. } (a_t'^{+,\sigma} - a_t^{*+,\sigma})^2 + (a_t'^{-,\sigma} - a_t^{*-,\sigma})^2 - Ft$$

S.t :

$$s_t^{\sigma} + a_t^{*-,\sigma} + a_t^{*+,\sigma} \geqslant 0, \forall \sigma \in \Sigma$$

$$s_t^{\sigma_c} \leqslant x^{\sigma_c}, \forall \sigma_c \in \Sigma_C \ ,$$

$$a_t^{*+,\sigma_p} \leqslant x^{\sigma_p}, \forall \sigma \in \Sigma_P \ ,$$

$$-\ a_t^{*-,\sigma_p} \leqslant x^{\sigma_p}, \forall \sigma \in \Sigma_P \ ,$$

$$\sum_{\psi \in \Psi} F_t^{\psi} \leqslant -d_t - \sum_{\sigma \in \Sigma} \eta^{\sigma} a_t^{*-,\sigma} + \frac{a_t^{*+,\sigma}}{\eta^{\sigma}} \ ,$$

$$\sum_{\psi \in \Psi} F_t^{\psi} \leqslant 0 \ ,$$

$$-\ F_t^{\psi} \leqslant c_t^{\psi} \ ,$$

$$-\ d_t' \leqslant d_t \ ,$$

$$-\ d_t' \leqslant 0 \ ,$$

$$\sum_{\sigma \in \Sigma} a_t^{*+,\sigma} \leqslant -d_t' - \sum_{\sigma \in \Sigma} a_t^{*-,\sigma} \ .$$

a' : Initial actions

a* : Fitted actions
(can be the same as initial)

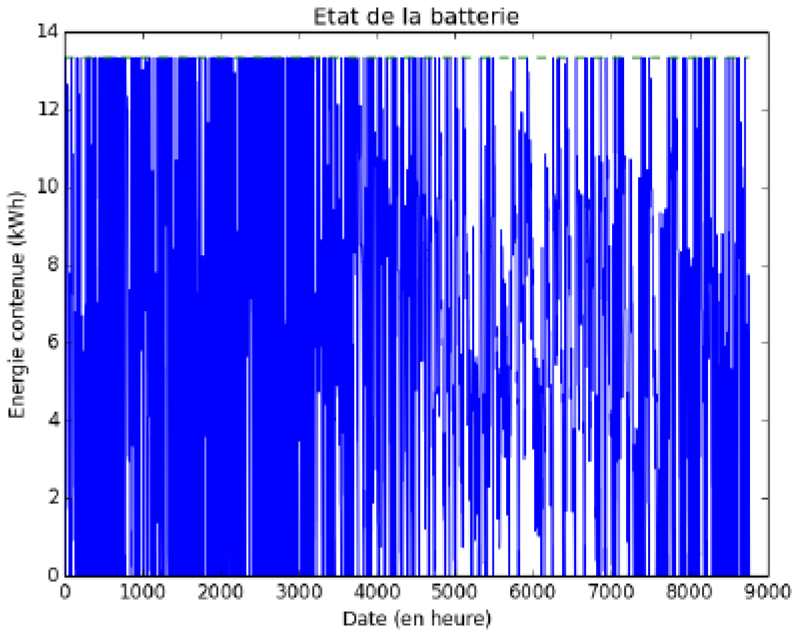
Constraints from linear program
(Slide 5)

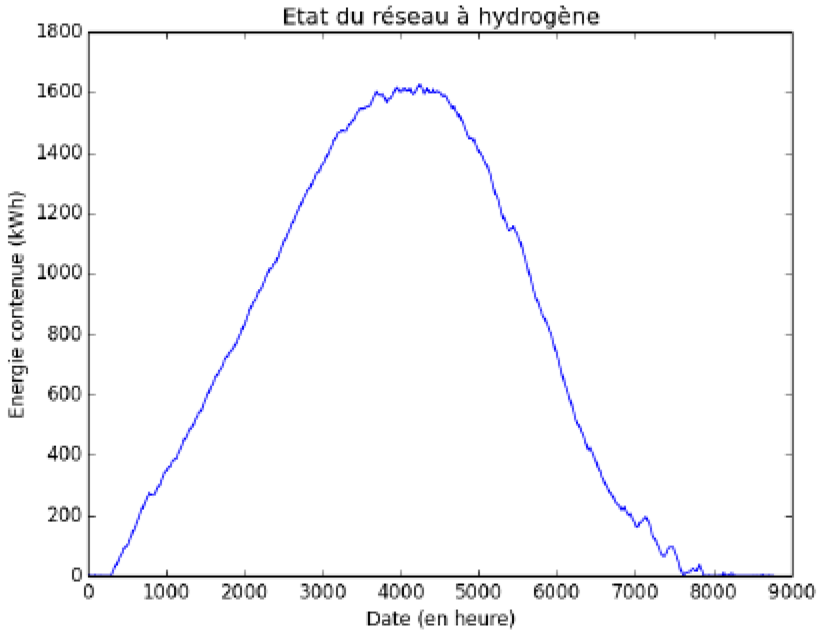
Additionnal constraints needed
because of different objective function

Test protocol

- Sequences of actions are sampled with solar production in Belgium and artificial load consumption (lack of real data...) computed by arbitrarily pattern.
- Several input spaces (from 12 hours up to 3 months of history) have been tested.
- The imitative scheduler will be compared to a greedy scheduler. The strategy of such a greedy scheduler consists in minimizing the energy wasting.

Optimal planning

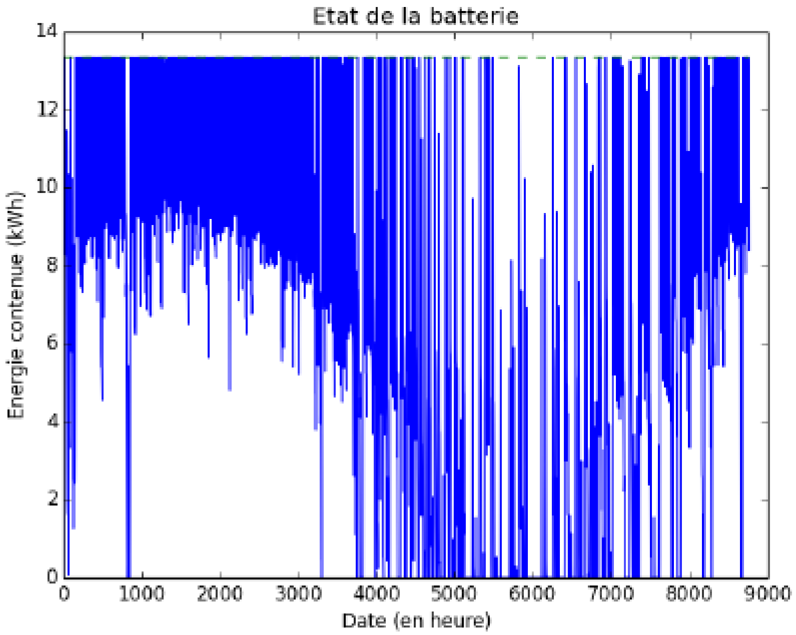


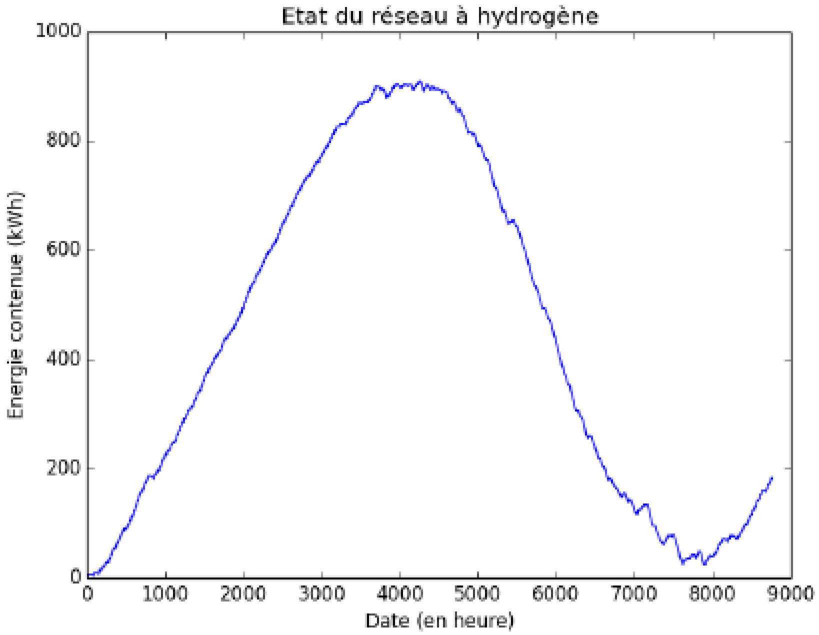


LEC : 0.32 € / kWh

Strong fluctuations of battery content.
Hydrogen tank content peak : 1800 kWh.

Greedy planning

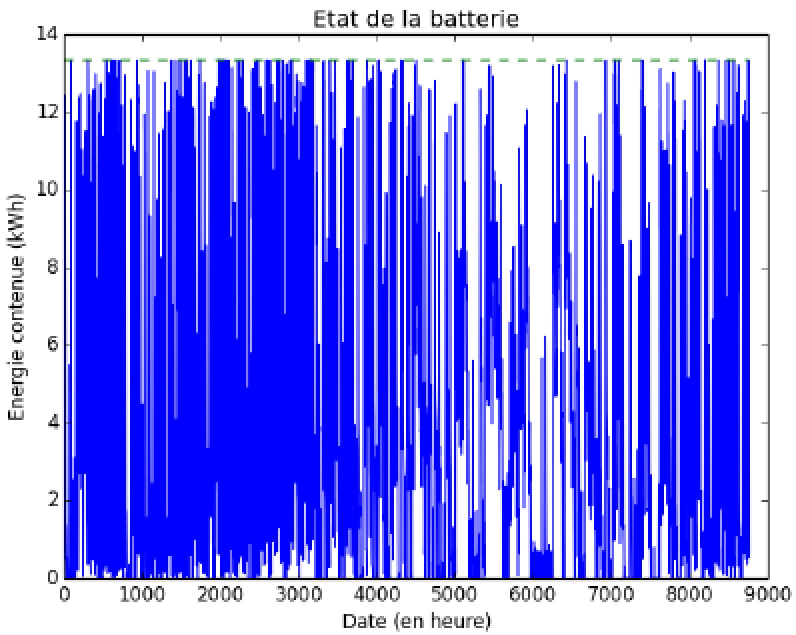


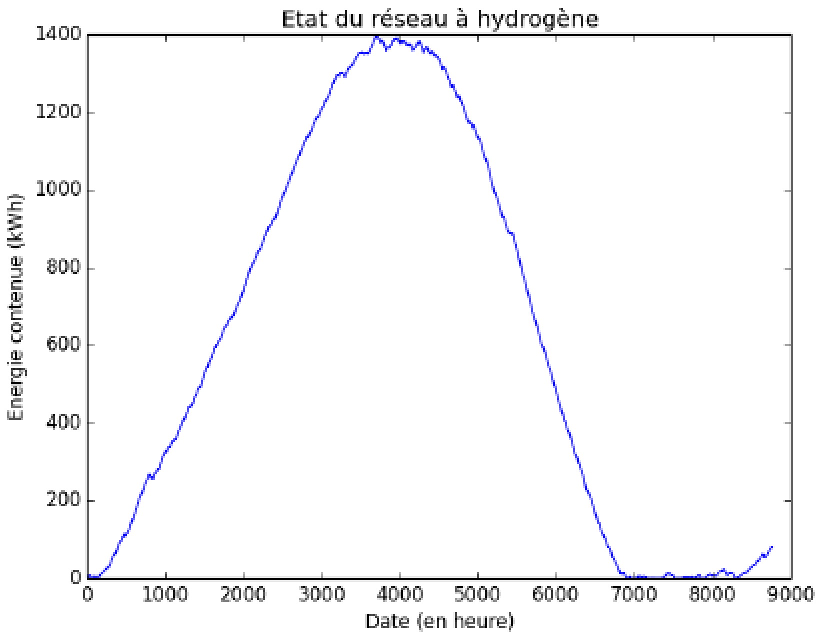


LEC : 0.6 € / kWh

Battery content not flushed at some points. Less energy wasting ?
Hydrogen tank content peak : 1000 kWh. Under-used.

Agent planning





LEC : 0.42 € / kWh

Battery content more often flushed than greedy.
Hydrogen tank content peak : 1400 kWh.
Still under-used but better than greedy.

Best input space found : 12 hours + additionnal attribute showing distance from current and summer equinox dates

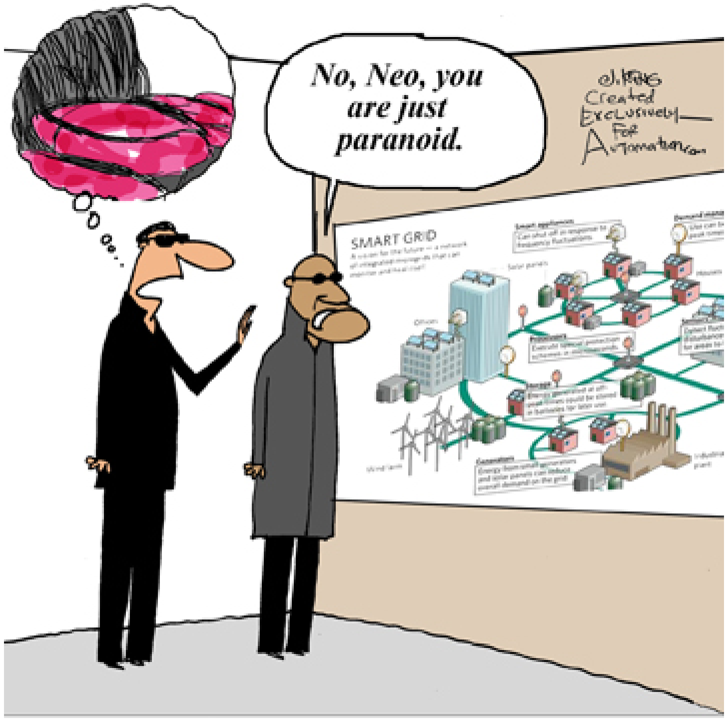
Conclusion

- Imitative learning scheduler performs better than the greedy scheduler.
- More data (realistic ones !) would further improve the approach.
- Future work :
  - Benchmarking on others learning structures ;
  - Developing a learning structure and his algorithm to take directly into account constraints ;
  - Addressing of online planning on more complex microgrids (non-linear dynamics, partial models...) using reinforcement learning.

References

- [1] Aittahar, S., Francois-Lavet, V., Lodeweyckx, S., Ernst, D. and Fonteneau, R. (2015) Imitative learning for online planning on microgrids. *In proceedings of Lecture Notes in Computer Science.*
- [2] Francois-Lavet, V., Gemine, Q., Ernst, D. and Fonteneau, R. (2015). Towards the minimization of the levelized energy costs of microgrids using both long-term and short-term storage devices. *Accepted*.
- [3] Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning, 63(1), 3-42*.

Thanks for your attention !



"Morpheus, I've heard this term, Smart Grid, before. It's what they called the Matrix before it took control of our lives."