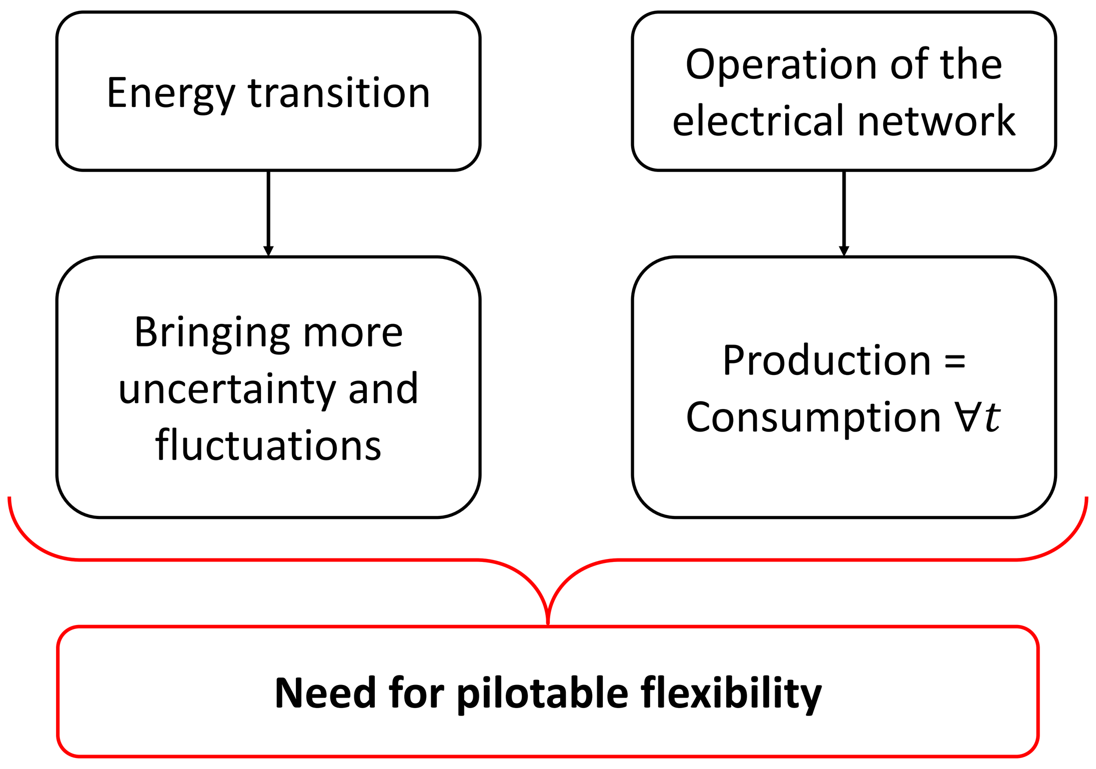
Machine Learning Informed Optimisation: Application to Pumped Hydro Energy Storage

Context

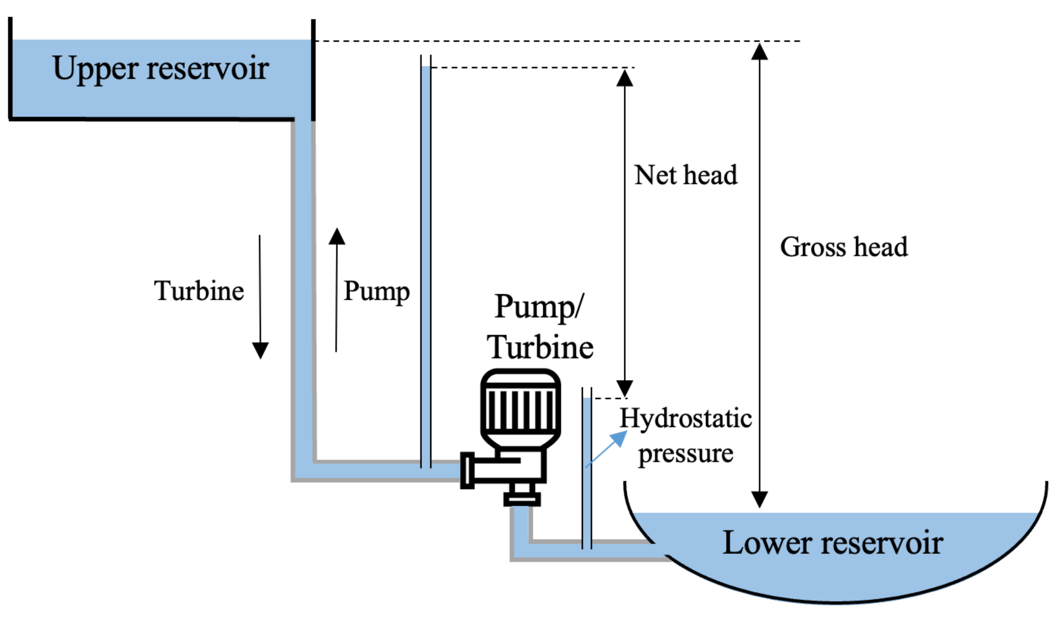
The increased contribution of uncertain and fluctuating renewable generation impacts the operation of power systems since the electricity production and consumption must be equal at all times.



Pumped Hydro Energy Storage

Storage brings flexibility since it can store energy when there is an excess of generation/a lack of consumption and, conversely, release electricity on the network when there is a lack of generation/an excess of consumption.

Pumped Hydro Energy Storage (PHES) uses water as a medium to store energy by pumping it to higher altitudes. This water can later be turbined to generate electricity. **Nowadays, 95% of storage capacity is PHES**.



Sketch of PHES

Work objectives

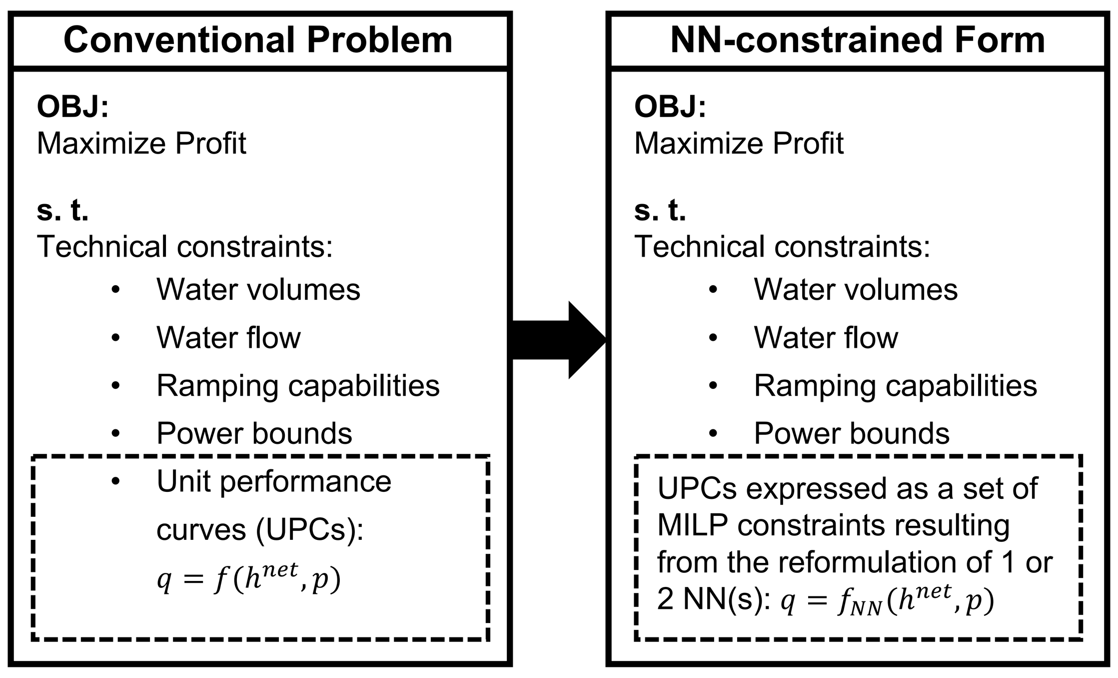
In order to decide which operations a PHES plant must perform; operators use models formulated as optimization problems. Ideally, those models must be convex, or MILP. However, the Unit Performance Curves (UPCs) (one per operating modes: pump or turbine) of the PHES plant are non-convex.

Therefore, **this work aims at leveraging the modelling power of neural networks to encode the operating curves of PHES systems.**

Day-ahead scheduling

The day-ahead scheduling is an optimization problem which aims at maximizing the profits of a plant on the day-ahead market. The operating schedule for the next day is obtained under constraints, including the UPCs.

In this work, the participation to the day-ahead energy-only and reserve markets is optimized jointly under a price-taker approach with perfect forecast.



Neural Networks

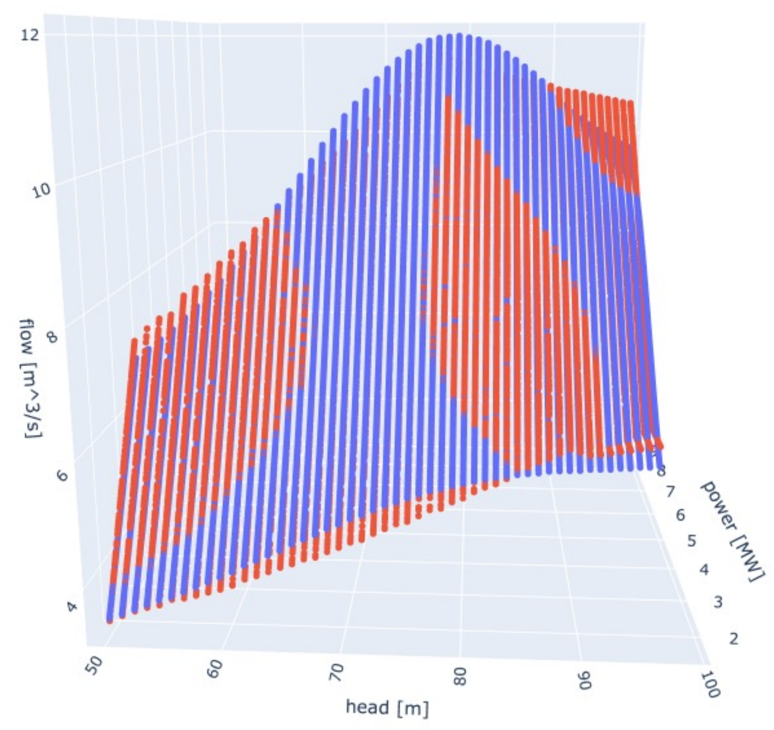
Neural Networks (NNs) are versatile modeling tools. **The complexity of the fit (and its quality) can be easily tailored by adjusting the number of neurons and layers.**

Any NNs with ReLU activation functions can be reformulated as a set of MILP constraints. This set is then embedded into the initial scheduling problem of the PHES plant.

$$
\begin{aligned}
& b_k \in \{0, 1\} \\
& y_k \leq \hat{y}_k - \hat{Y}_k^{\min} \cdot (1 - b_k) \\
& y_k \geq \hat{y}_k \\
& y_k \leq \hat{Y}_k^{\max} \cdot b_k \\
& y_k \geq 0
\end{aligned}
$$

ReLU formulation where $\hat{y}_k$ is the input of the neuron, $\hat{Y}_k^{\min}$ and $\hat{Y}_k^{\max}$ are the input bounds, y_k is the output.

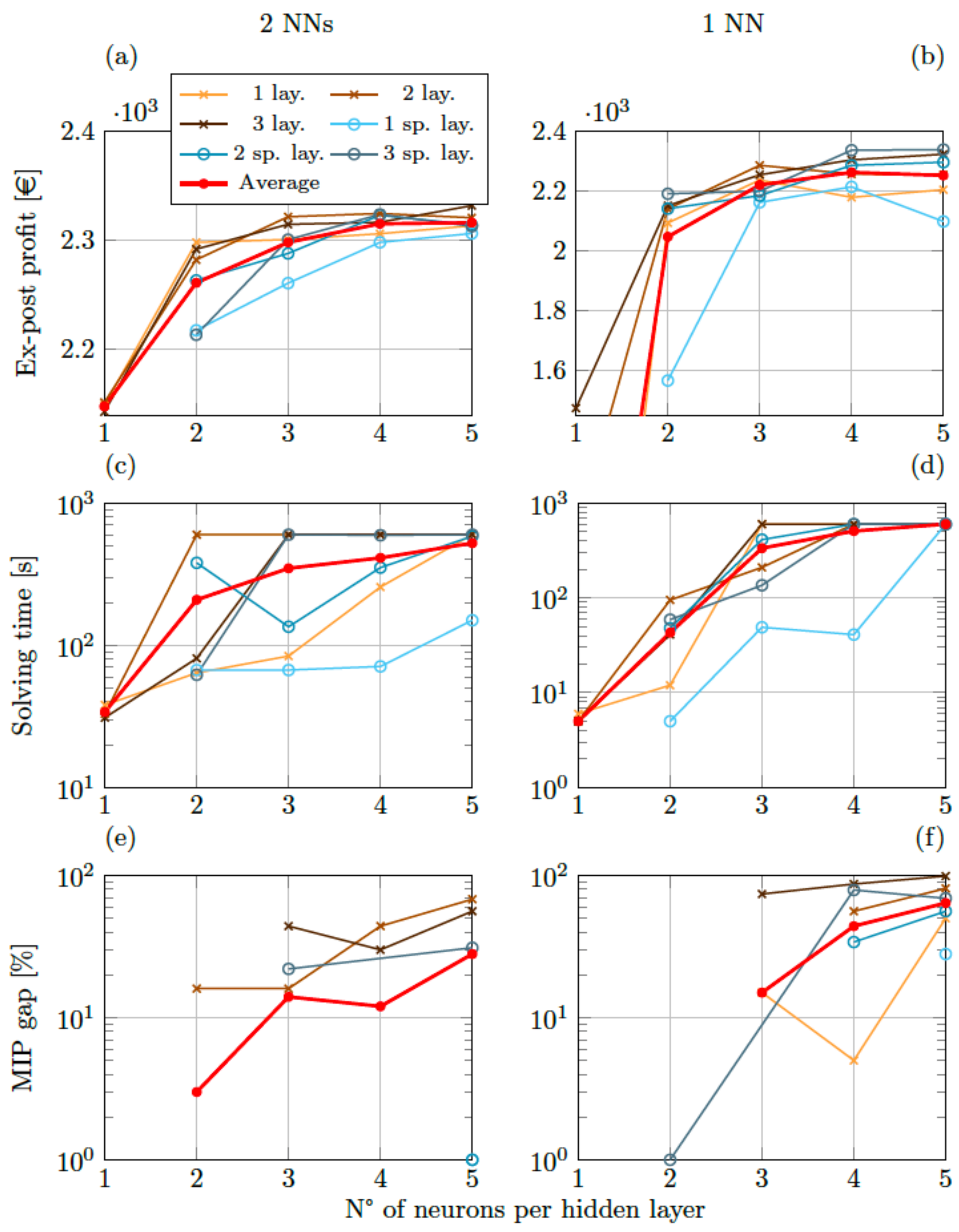
One NN can be used per UPC to be modelled (see below for the turbine) or for both turbine and pump UPCs.



Original turbine UPC
NN approx. (1 hidden layer with 2 ReLU neurons)

Results

The different dispatch performances are compared over a typical day. A detailed PHES simulator, mimicking the minute-wise PHES behavior, is developed to accurately assess the feasibility and economic performance of the resulting schedules.



The **ex-post profit increases with the number of neurons** per layer and the number of layers. The solving time follows a similar trend.

Architectures featuring weight pruning (in blue shades) present quicker solving time and competitive ex-post profits, sometimes even outperforming their conventional counterpart.

Overall, the 2NN approach has higher ex-post profits with similar solving times to the 1NN approach.

Conclusion

- **NNs are a very versatile tool to model non-linear curves and can be reformulated into a MILP problem.**
- The solving time increases quickly but weight sparsity allows to reduce it.
- The tuning of the hyperparameters (architecture, weight pruning rate, etc.) is challenging.
- Look into other reformulations of the activation function
- Use other piecewise functions such as Leaky ReLU

Université de Mons
Pietro Favaro (pietro.favaro@umons.ac.be)
under the supervizion of François Vallée and Jean-François Toubeau