POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Scalable Power Network Control with Reinforcement Learning

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Daniele Paletti

Advisor: Prof. Marcello Restelli

Co-advisor: Alberto Maria Metelli

Academic year: 2021-2022

1. Introduction

We are in the middle of a climate crisis, renewable energies must readily be implemented in the power grids. However, solar and wind power integration in the grid is challenging, their production depends on the weather, and our overall storage capacity is insufficient. We hope artificial intelligence (AI) can assist grid operators.
For this reason, RTE (Réeseau de Transport d'Electricité) has been organizing the **"Learning to Run a Power Network" (L2RPN)** challenge to foster AI applications to power network control, refer to fig. 1 for a timeline of the competitions. L2RPN challenges cast the power network control problem in the Reinforcement Learning (RL) framework. The actions allowed to the autonomous agent are akin to those available to a human operator (e.g. line switching and power production changes). The control problem ends when the total demand is not met anymore, i.e., a blackout is triggered.
Challenges offer standardized and reproducible benchmarks that alleviate the AI reproducibility crisis. However, scientific competitions force participants to build instance-optimized solutions with limited real-world applicability. We developed a model to solve the challenge independently of any competition and used the L2RPN testbed for evaluation.
We developed a hierarchical Multi-Agent RL (MARL) system:

- **Multi-Agent**: multiple agents participate in each decision;
- **Hierarchical**: managers handle communities of agents and select the best decision proposed by each community.

We evaluate our model on two environments of increasing size and complexity. Finally, we show that the model performs substantially better than a challenging expert system proposed by the L2RPN organizers in both settings.

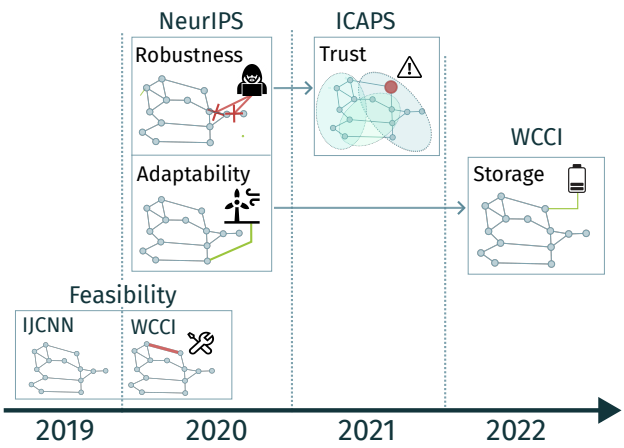


Figure 1: L2RPN timeline.

2. Background

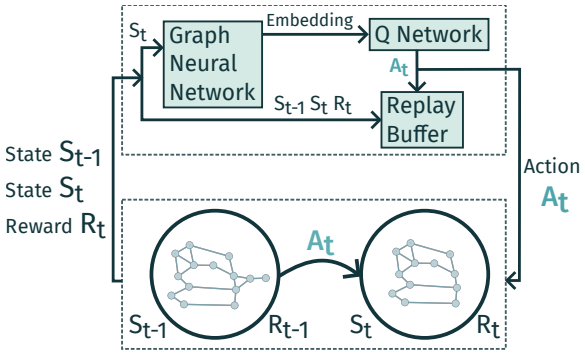


Figure 3: System's agent overview.

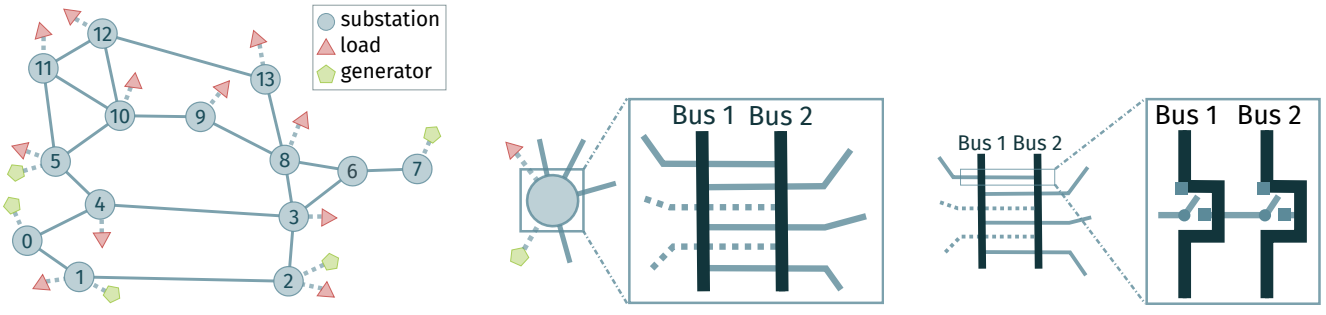
A RL agent must be able to sense the state of its environment and must be able to take actions to affect the state and reach a specific goal encoded through a reward (see fig. 3). State-action pairs are evaluated through the action value function Q. Identifying the correct value for each state-action pair equals to solving the RL task. State-action spaces are often huge, thus the Q-function needs to be estimated. All the actors in the system are Double Dueling Deep Q-Networks with Prioritized Experience Replay:

- **Q-learning**: an off-policy temporal-difference (TD) control algorithm that approximates Q; TD methods are a combination between Monte Carlo and Dynamic Programming;
- **Deep Q-Network**: an agent which combines Q-learning with deep neural networks to approximate the value of each state-action pair;
- **Dueling Networks**: neural network architectures suited for RL that separate the representation of state and action values;
- **Prioritized Experience Replay**: the experience replay buffer stores agent experiences, and the agent uniformly samples experience batches during learning. Prioritization introduces importance sampling to prioritize the most important experiences.

All the actors perceive their environment through **Graph Neural Networks** (GNNs). The GNN formalism is a general framework for defining deep neural networks on graph data. We generate representations of nodes that depend on the graph's topology and features.

3. Problem Description

The L2RPN challenge is a series of competitions that model the sequential decision-makig environments of real-time power network operation. The participants' algorithms must control a simulated power network within an RL framework. RTE has developed Grid2Op, a python module that casts the power grid operational decision process into a Markov Decision Process (MDP). Grid2Op represents a power grid as a set of objects: powerlines, loads, and generators, with substations linking everything together. **Powerlines connect substations and allow power to flow from one place to another**. A graph akin to the one in fig. 2a naturally models the power grid as we have described it. However, our power network model needs to take into account also the **internal structure of substations**, where we find two busbars to which every grid object connects (see fig. 2b). Substations connect elements through switches which allow the operator to electrically separate elements from each other (see fig. 2c).



(a) L2RPN Environment. (b) Substation Buses. (c) Bus Switches.

Figure 2: Power Network Model.

4. Related Works

Many practitioners apply RL to power network control. Refer to table 1 for an overview of RL in the power network literature.

Several authors developed MARL systems to build controllers capable of dealing with the stochasticity induced by renewable generators while scaling to large topologies. Xi et al. [3] designed a cooperative game among network agents and solved it through Nash Q-learning. Zhang et al. [4] employed a decentralized consensus algorithm with agents managing predefined network areas.

We extend Zhang's et al. idea by having managers deal with **evolving communities rather than fixed network areas**. The power network evolves through time, and every fixed clustering sooner or later becomes outdated. Then, we take Xi's et al. decision structure and make it **hierarchical rather than horizontal**. A purely horizontal decision structure hinders scalability because consensus speed depends on the number of agents reaching agreement.

5. Solution

We developed a **hierarchical multi-agent RL system** (see fig. 4). The system has three main actors: substation agents, community managers, and head managers.

Every substation houses an agent which perceives only its immediate neighborhood, i.e., the directly connected powerlines and substations. Each agent takes one action per timestep given its current immediate neighborhood and experiences. Every substation agent may handle any number of buses. In the Grid2Op model, each substation houses two buses.

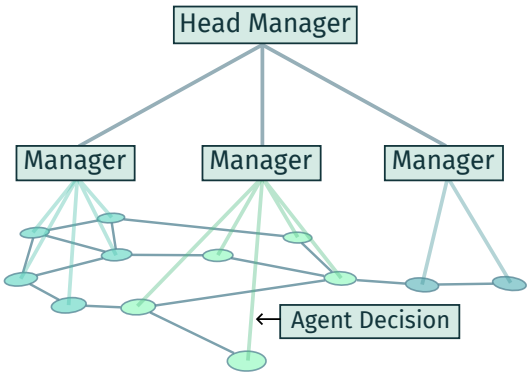


Figure 4: Solution architecture.

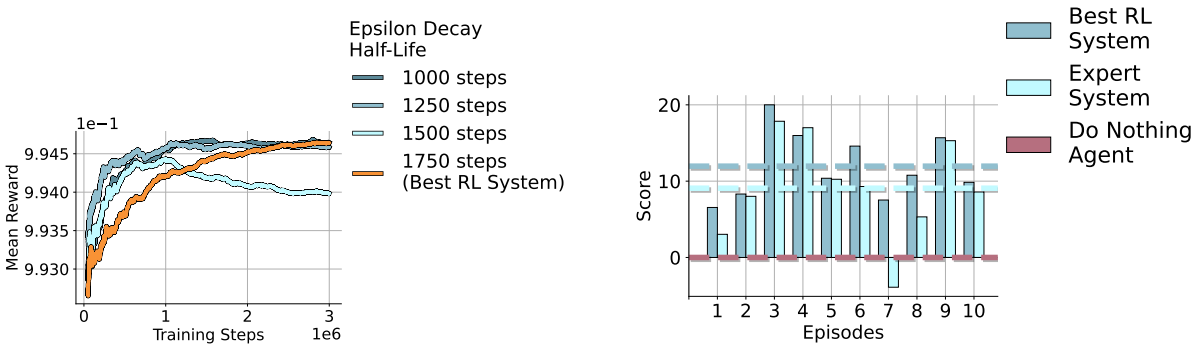
The system then builds agent communities: subgraphs composed by reducing the number of outgoing edges and maximizing inner edges. We detect communities dynamically through DynaMo [5], an extension of the Louvain algorithm to dynamic networks. **Each community manager handles a community of agents**. Each manager selects an agent belonging to the community, given the community structure, all the current agents' actions, and their experience.

Finally, **the head manager receives all the managers' choices and must choose one**. Then, the action of the chosen agent gets executed. The head manager picks a manager given a summarized version of the graph where communities are represented as nodes linked by inter-community edges. Like all other system actors, the head manager is an RL agent, thus pairing its current perception with experience to make a decision.

RL Advantages	RL Disadvantages
▷ Leverage incomplete information	▷ No consideration for devices' physical structure
▷ Learn continuous control under continuous state-action spaces	▷ Lacking of multi-timescale decisions
▷ Deal with unpredictable emergencies	▷ Generator production control is not coupled with voltage and frequency control

RL Directions
▷ Combination of RL and classical control methods
▷ Hierarchical strategies layering control and optimization
▷ Merging of grid data features with device model features

Table 1: RL in the power network literature.



(a) Training (b) Evaluation

Figure 5: NeurIPS 2020 environment experiments

6. Results

6.1. Experimental Setup

We paired our RL system with the rule-based **Expert System (ES)** adopted in the WCCI 2022 baseline [1].

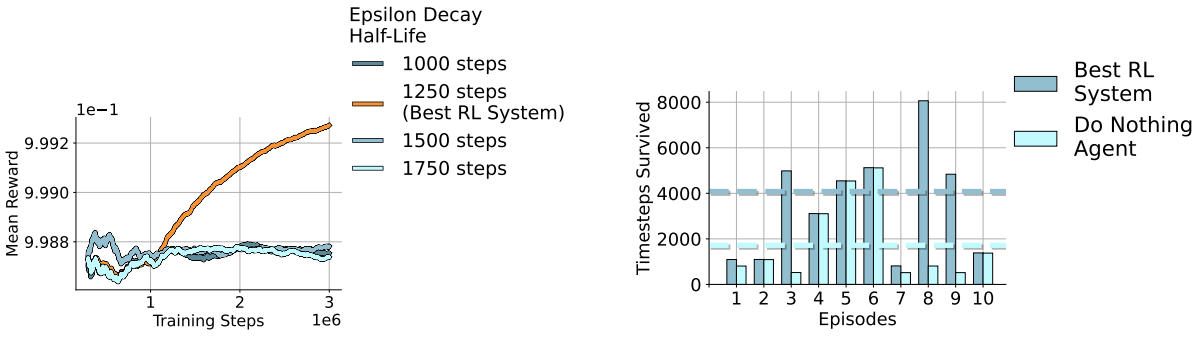
The **reward** is 1 if no blackout happens, -1 otherwise and the episode terminates.

The agent is evaluated using the WCCI 2022 **score**, which estimates the power network operational cost scaled between 0 and 100, where 0 is achieved by the Do Nothing Agent (DNA).

We adopt the **ϵ-greedy** exploration strategy with exponentially decaying ϵ: the agent takes a random action with probability ϵ otherwise the action that maximizes the Q-function in the current state.

6.2. Experiments

In fig. 6, we show the experiments on the **L2RPN case 14 environment**, a relatively small environment counting 14 substations. Figure 6a shows the reward achieved with respect to different values of the ϵ-decay half-life, set equal for all systems' actors. The network size induces many local maxima. Thus, the system is susceptible to such hyperparameter. In Figure 6b, we show the timesteps survived on 10 test episodes by the best controller and the DNA. In this environment, the ES and the DNA score are the same. **The best RL controller achieves a mean score of 35 in a 30-episodes test set**. In fig. 5a, we repeated the tests we conducted on the case 14 environment on the **NeurIPS 2020 Environment**, which counts 36 substations, has a random attacker cutting the powerlines, and planned maintenance periodically disables some powerlines. The increased complexity of the control task reduces the sensibility to ϵ. In fig. 5b, we show the score of the RL agent with respect to the ES alone: **the RL agent scores 10.09 in a 30-episodes test set, 2.7 points higher than the ES alone**. The ES alone scores 7.39 points over 30 test episodes



(a) Training (b) Evaluation

Figure 6: Case14 environment experiments

because it deals with some of the complexities added by the attacker and the maintenance.

7. Conclusions

The advent of **renewable energy** on the production side poses significant challenges to grid operators. The power system community has been lately focusing on deep RL solutions for their capacity to learn representations and parallelizable architectures. For this reason, RTE instituted the L2RPN challenges: a series of competitions that model the sequential decision-making environments of real-time power network operation.

On the one hand, several authors have focused on MARL systems to deal with the size of real-world power networks. On the other hand, the challenge format asks participants to build ad-hoc solutions incentivizing single-agent models.

All this considered, we decided to build a **hierarchical MARL system** taking advantage of the L2RPN ecosystem without participating in any competition. The hierarchical structure factors the action space among several agents yielding a **scalable architecture**. Each substation agent perceives its immediate neighborhood; community managers perceive evolving agent communities, and the head manager perceives a summarization of the original power network.

We evaluated our system on the L2RPN case 14 environment and the NeurIPS 2020 environment using the same scoring adopted in the WCCI 2022 challenge. On the L2RPN case 14 environment, we scored 35/100 points above the Do Nothing Agent. On the NeurIPS 2020 environment, we scored 10.09/100 points above the Do Nothing Agent and 2.7/100 points above a challenging expert system.

The experimental results demonstrate the **feasibility** of our approach and show **non-negligible performance improvements** over the benchmarks.

Finally, we showed how scientific challenges could be meaningful, standardized, and reproducible benchmarks rather than mere competitions.

8. Future Work

8.1. Intrinsically Motivated RL

Throughout all the experiments, we used the same exploration strategy for all system actors. However, every actor perceives a different subsection of the environment, thus needing a specific degree of exploration. Intrinsically motivated RL may improve actors' exploration strategies by driving the **exploration policy based on each actor's perception**.

8.2. Pointer Networks

Managers deal with variable inputs as communities evolve and the power network changes. In our design, managers use masking to deal with such variability. However, the preferences of each manager must be expressed over all network substations and then masked to keep only that of the current handled community. This approach may yield performance degradation over larger topologies.

We propose implementing the manager decision layer with Pointer Networks (PNs). Such networks allow the selection of the best element from a **variable-sized input sequence without depending on sequence length**. On top of this, Graph PNs and Hybrid PNs show successful applications to graph data.

8.3. Power Supply Modularity

We detect communities with DynaMo, an extension of the Louvain algorithm to dynamic networks. Such algorithms take into account only topological features. This assumption is frequently reasonable for graph data. However, power networks yield much more information.

To tackle this issue, we propose to adopt Power Supply Modularity (PSM) [2] as an alternative measure to **replace the Louvain modularity**. The PSM considers the complex electrical properties and the functionality of power networks yielding more accurate community detection.

References

[1] Antoine Marot, Benjamin Donnot, Karim Chaouache, Adrian Kelly, Qiuhua Huang, Ramij-Raja Hossain, and Jochen L. Cremer. Learning to run a power network with trust, April 2022.

[2] Xiaoliang Wang, Fei Xue, Shaofeng Lu, Lin Jiang, Ettore Bompard, and Marcelo Masera. Understanding Communities From a New Functional Perspective in Power Grids. *IEEE Systems Journal*, 16(2):3072–3083, June 2022.

[3] Lei Xi, Jianfeng Chen, Yuehua Huang, Yanchun Xu, Lang Liu, Yimin Zhou, and Yudan Li. Smart generation control based on multi-agent reinforcement learning with the idea of the time tunnel. *Energy*, 153:977–987, June 2018.

[4] Xiao Shun Zhang, Qing Li, Tao Yu, and Bo Yang. Consensus Transfer Q - Learning for Decentralized Generation Command Dispatch Based on Virtual Generation Tribe. *IEEE Transactions on Smart Grid*, 9(3):2152–2165, May 2018.

[5] Di Zhuang, J. Morris Chang, and Mingchen Li. DynaMo: Dynamic Community Detection by Incrementally Maximizing Modularity. *IEEE Transactions on Knowledge and Data Engineering*, 33(5):1934–1945, May 2021.