POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Graph-Based Multi-Agent Reinforcement Learning for Power Grid Control

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Carlo Fabrizio

Advisor: Prof. Marcello Restelli

Co-advisors: Marco Mussi, Gianvito Losapio, Alberto Maria Metelli

Academic year: 2024-2025

1. Introduction

The proliferation of renewable energy sources and the electrification of large, energy-intensive sectors, are driving significant changes in the structure and complexity of modern power grids. The variability associated with renewable energy sources, whose production heavily depends on the atmospheric conditions, and the exploding size of power networks, counting a huge number of potential configurations, prevent traditional control systems from being effective. To properly cope with these obstacles, modern power grids require novel and adaptive control strategies, that are scalable and capable of ensuring real-time responsiveness. In order to investigate the potential use of Artificial Intelligence (AI) within the next-generation power grid controllers, the French electricity network management company RTE (Réseau de Transport d'Électricité) developed a series of "Learning to Run a Power Network"(L2RPN) challenges [4]. These challenges, offered over the Grid2Op framework, were designed to simulate realistic sequential decision-makings environments in which agents must keep the power grid operational under different conditions of uncertainty, by employing a wide range of control actions and interventions. The last challenge ended in 2023, but the research activity in the context of next-generation power grid controllers continued beyond the scope of the challenges. Current research is pushing two main directions: implementing distributed multi-agent solutions to decompose the vast action space, and using graph-based learning models to incorporate network information into the decision-making process. Several Multi-Agent distributed solutions have been proposed. The architecture design, that proved to be the most effective, consists in a network of low-level agents, directly operating on power grid's components, coordinated by a high-level manager responsible for selecting which agent should intervene in response to danger situations. Regarding the use of graph-based learning modules, no standard framework has yet emerged as the most effective or widely adopted by the research community. Many solutions employ Graph Neural Networks (GNNs) to learn on graph topologies. Most of them represent the power network as a graph where nodes identify various types of power grid elements [2]. However, such heterogeneous graph representation exhibits severe limitations in expressiveness, making it unsuitable for the task.

A complementary line of research explores methods for decomposing the power grid into smaller, independent subgrids to be managed separately. This approach aims to reduce the overall complexity of the control problem and enable the development of more scalable and modular solutions. Current proposed approaches compute a static decomposition of the power grid, which is, instead, inherently a dynamic and evolving system, potentially exhibiting non-fixed subgrids over time. In light of the research trends discussed above, this thesis aims to go one step further. It proposes a distributed Multi-Agent Reinforcement Learning (MARL) solution that integrates a GNN, **enabling the decomposition of both the action and observation spaces**. The proposed framework splits the action space among several independent agents and leverages the GNN to generate informative local observations, entirely addressing the scalability challenges. A side contribution involves the implementation of a dynamic and learnable clustering procedure for substations, aimed at enabling adaptive decomposition of the power grid based on observed behaviors.

2. Problem Formulation

The work was conducted over the Grid2Op simulation environment, which is designed to simulate realistic power grids for a given period, usually several days at 5-minutes intervals. At each timestep, the agent is called to take an action on the simulator, leading to the next simulator's state. The power grid is represented as a graph of connected elements: nodes and edges represents respectively substations and powerlines. The simulator also models other elements, specifically, generators and loads. Generators produce electricity while loads consume it. Substations can be viewed as routers within the power network: they encompass two internal buses to which their elements (loads, generators or powerlines) can be connected. Substations have no other role than controlling their internal connections affecting the overall flow of energy. Among the various interventions supported by the simulator, topology-based actions are undoubtedly the most interesting and extensively studied. This is because they have no operational cost and are highly complex due to their combinatorial nature, making them particularly challenging for humans to execute effectively. Consequently, this thesis focuses exclusively on topology-based actions. These actions allow modifications to the internal connectivity within substations. Thus, a substation i controlling N_i elements can perform up to 2^{N_i} actions, corresponding to all possible configurations of its elements across the two buses. To ensure system stability, Grid2Op allows only one substation to be controlled at each time step, as simultaneous interventions may cause unpredictable interactions and instabilities. The simulation runs for at most T timesteps, but it ends prematurely in case of a blackout. A blackout can occur if the energy demand (loads request) is not satisfied by the current configuration. The primary cause of a blackout is known as "line overload". When the current flow of a specific powerline exceeds a threshold, the simulator automatically disconnects the powerline. This can potentially lead to a grid configuration that violates the load requirements. Grid2Op offers a large number of features at each time step to describe the environment's state, more details can be found in the official Grid2Op documentation. The objective of a control algorithm is to keep the powergrid stable despite the unknown evolution of energy production and consumption. The performance metric, known as "survive time," corresponds to the number of time steps the grid remains operational before a blackout occurs during an episode. Taking into account the several rules and features of the simulator, the power grid control problem can be formalized as a Markov Decision Process (MDP). Notably, as the size of the power grid grows, the action space grows with the number of controllable elements. Moreover, the state representation becomes hard to be learned effectively due to the expanding grid's graph structure and increasing number of complex interconnections. An efficient control algorithm must therefore handle an extremely large action space while leveraging graph-structured data to generate informative observations.

3. Proposed Solution

The proposed solution is a Graph-based Multi-Agent RL algorithm for real-time scalable power grid control. The method consists of a network of distributed low-level agents, each controlling a different powerline, coordinated by a high-

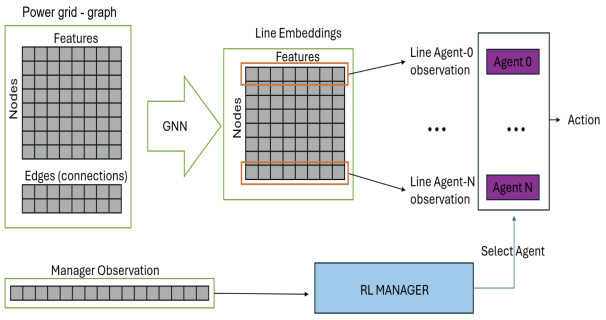

Figure 1: Overview of the proposed approach.

level manager. Each low-level agent operates solely on its associated power line, working on an action space including only the configurations of its two extremity substations. Additionally, each low-level agent receives a local observation preprocessed by a shared GNN, which provides information about the neighboring power lines. **A novel representation models the power grid as a homogeneous graph in which powerlines are nodes and substations are edges**. To better illustrate how the environment's state s_t is processed to obtain the observation of a single agent i, the procedure is defined by the following equations:

$$g_t = \text{convert\_graph}(s_t) \tag{1}$$

$$o_t^i = \left[\text{GNN}(g_t \mid \phi^t)\right]_i , \tag{2}$$

where $[\dots]_i$ is the i-th element of the input vector. The proposed procedure decomposes the action space and the observation space simultaneously, enabling a fully scalable solution. An abstract overview of the method is provided in Fig. 1. To further improve learning efficiency and stability, the framework incorporates Deep Q-Learning from Demonstrations (DQfD) [3] and Bootstrapped potential-based Reward Shaping (BSRS) [1]. DQfD enables agents to learn from a dataset of expert demonstrations in an offline manner, thereby enhancing training stability and accelerating convergence. BSRS is employed to implement a sort of credit assignment in a scenario in which the global reward is highly diluted. This allows each agent to learn from a more informative reward signal, potentially improving convergence in large-scale power grid environments. Low-level agents implement BSRS; thus, each low-level agent i

Algorithm 1 Training Procedure

1: {**Pre-training phase**}
2: All agents learn from expert demonstrations
3: {**Online training phase**}
4: **for** each training iteration **do**
5: Agents interact with the environment and store new experiences
6: Low-level agents and the GNN jointly update the policies using BSRS
7: High-level manager updates its policy
8: **end for**

learns from the shaped reward:

$$\tilde{r}_t^i = r_t + \gamma\eta\, V_i^n(s_{t+1}) - \eta\, V_i^n(s_t) \tag{3}$$

where $V_i^n(\cdot)$ denotes the agent's estimate of the state-value function at time n. An overview about the training procedure is provided in Alg. 1.

All the agents, low-level and manager, are Deep Dueling Double Q-Learners (DDDQN) equipped with a Prioritized Experience Replay Buffer (PERB). The GNN is a Graph Attention Network v2 (GATv2) and receives as input the line-graph representing the power grid. As previously stated, low-level agent i receives as input the i-th component of the output of the GNN, thus, the actual target computation for low-level agent i, during the online phase, is:

$$\begin{aligned} y_t^i = \tilde{r}_t^i + \gamma \cdot Q\Big(&g_{t+1}, \\ &\arg\max_a Q(g_{t+1}, a | \phi^t, \theta_i^t, \alpha_i^t, \beta_i^t) \mid \phi^{t-}, \theta_i^{t-}, \\ &\alpha_i^{t-}, \beta_i^{t-})\Big), \end{aligned} \tag{4}$$

where ϕ represents the learnable parameters of the GNN and are shared among all the agents (no subscript i), θ_i, α_i and β_i are the Dueling Q-network parameters, respectively: the shared ones, the ones of the advantage stream and the ones of the value stream. The superscript $^{-t}$ indicates frozen parameters from the target network. To better illustrates the computational steps required to produce an action, Alg. 2 outlines the flow executed by the architecture when called to act on the environment.

The expert experiences for DQfD come from an Expert algorithm that makes informed, domain-specific decisions by means of a large number

Algorithm 2 G4DQN computational flow

if Danger situation **then**
  $i \leftarrow \text{Manager.select\_agent}(s_t|\mu^t)$
  $g_t \leftarrow \text{convert\_graph}(s_t)$
  $o_t \leftarrow \text{GNN}(g_t|\phi^t)[i]$
  $a_t \leftarrow \text{Agent}_i\text{.policy\_play}(o_t|\theta_i^t, \alpha_i^t, \beta_i^t)$
  return a_t
end if

Algorithm 3 High-Level rule-based logic

if No line disconnected and no line overloaded **then**
  {Safe situation}
  Do nothing
else
  if There is a disconnected line **then**
    {Critical situation}
    Reconnect the line
  else
    {Danger situation}
    Let the agents play
  end if
end if

of simulations. On top of the architecture described above, a top-level rule-based logic and an action space reduction technique are employed. These two components have proven to be essential for designing competitive power grid controllers, as they are included in most of the winning solutions of the L2RPN challenges [5]. The former, responsible for determining when the RL agent should intervene, is well detailed in Alg. 3. The latter, instead, consists in a single-step simulation performed at runtime to eliminate unsafe actions. Before a low-level agent executes an action, the actions at its disposal are simulated for a single step, and only those that do not worsen the grid's stability are retained. Although it increases the model's inference time, the action space reduction technique is necessary to cope with the significant computational time required to achieve convergence within a single experiment in the Grid2Op simulator.

4. Experimental Results

The experiments were conducted on the Grid2Op simulator, in particular on the *"l2rpn_case14_sandbox"* environment. The latter represents a power grid counting 14 substations, 20 lines, 6 generators and 11 loads, and it encompasses 1004 different episodes (also referred to as chronics). The chronics last for a maximum of 8064 timesteps. If the agent survives until the end, the episode terminates regardless of whether a blackout has occurred. The proposed algorithm controls each powerline by means of a dedicated low-level agent, resulting in 20 such agents coexisting in the same environment. Including the RL manager, the system consists of 21 agents in total. Each agent maintains two neural networks, a main network and a target one. In addition, the GNN with both main and target versions is also maintained. In total, 44 neural networks are simultaneously loaded on the computing device, with 22 actively trained and the remaining 22 kept frozen for target computation. Due to the complexity of the training setup, gradient clipping and soft main-target synchronization were adopted to achieve stable learning and effective training. To properly evaluate the model's performance, the 1004 available chronics were divided into 984 for training (98%), 10 for validation (1%), and 10 for testing (1%). The proposed algorithm was compared against a straightforward yet surprisingly competitive baseline: the *Do-Nothing* policy, the same baseline used in L2RPN challenges. The results over the 10 test chronics are provided in Fig. 2, which show that the proposed model outperforms the baseline. For a deeper understanding of the model's components, an ablation study was conducted. The contribution of the components of the proposed architecture was analyzed by removing them one at a time and observing the impact on performance. A summary of the results is provided in Tab. 1. The ablation study highlights that components such

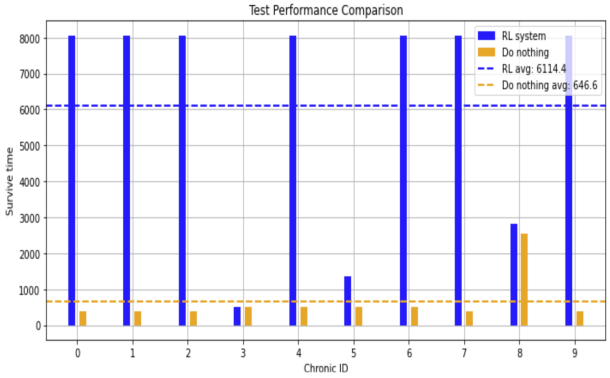

Figure 2: Proposed model vs *Do Nothing*.

Table 1: Ablation study results: average survival time on validation and test sets. The best result per column is in bold, the second-best is underlined. The first raw – Do Nothing – reports the performance of the baseline.

Configuration	Val Performance	Test Performance
Do Nothing	997.7	646.6
No DQfD	1985	1877.9
No GNN	1206.4	785.2
No RL Manager (CAPA)	2562.5	—
No Reward Shaping	**5667.1**	<u>5324.3</u>
Complete System	<u>5452.2</u>	**6114.4**

Table 2: Inference Time Comparison: Proposed approach against Expert agent.

Model	Inference Time (avg ± std) [s]
Proposed Model	0.187 ± 0.145
Expert Agent	2.56 ± 0.223

as DQfD, the RL manager, and the GNN are essential to achieve competitive performance. In contrast, the impact of BSRS is less evident in the current environment. However, it is reasonable to expect that in larger and more complex grid topologies, where reward signals may be highly diluted, BSRS could play a more critical role in stabilizing and accelerating the learning process. To analyze the efficiency of the model, the computational requirements of the proposed approach were investigated. In general, simulating a power grid is, by nature, a computationally intensive task. The plain power grid simulation takes on average 0.1097 seconds to simulate a single step. Thus, it takes almost 15 minutes to simulate all the 8064 timesteps of an episode. The inference time of the proposed approach is compared against the one of the Expert agent used to collect expert demonstrations, and the results are provided in Tab. 2.
The results presented in this section show that the proposed approach is a competitive solution, outperforming the Do-Nothing baseline. Moreover, it proves to be significantly more computationally efficient than the Expert Agent, demonstrating both its scalability and effectiveness. The complete implementation is available at: https://github.com/Carlo000ml/RL4PG.

5. Power Grid Decomposition

A side contribution was given towards the implementation of a technique for the dynamic decomposition of a power grid into subgrids. The

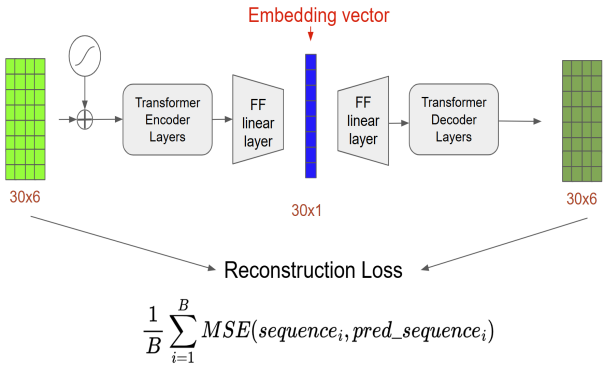

Figure 3: Transformer Autoencoder Architecture. It receives in input a sequence of shape (30,6), compress it into shape (30,1) and then reconstruct the original sequence of shape (30,6). It is trained to minimize the Mean Squared Error (MSE) between the input sequence and the output one.

goal was to develop an adaptive clustering of substations, i.e., a clustering that evolves over time as the topology of the power grid changes. The procedure begins by modeling each substation as a time series. Specifically, the buses' sequences of power flow values, observed during an episode, are used to characterize the substation's behavior. This temporal representation captures how the substation operates over time, enabling the identification of patterns or similarities across different substations. These time series are then compressed using a Transformer Autoencoder, previously trained to reconstruct them by means of an encoder-decoder structure. An overview of the Transformer Autoencoder architecture is shown in Fig. 3.
The compressed data points are finally clustered using in parallel Hierarchical Clustering (HC) and DBSCAN, in order to ensure consistency of the results. The overall procedure is illustrated in Fig. 4.
The method was tested on time series generated by running a random policy over the test set chronics, which were not used for training or validating the autoencoder. Both HC and DBSCAN suggest that, for the majority of the chronics, two is the optimal number of clusters. Although DBSCAN identified minimally different clusterings in 4 out of the 10 chronics, the remaining episodes consistently exhibited the same clustering when using either HC or DBSCAN, which is shown in Fig. 5.

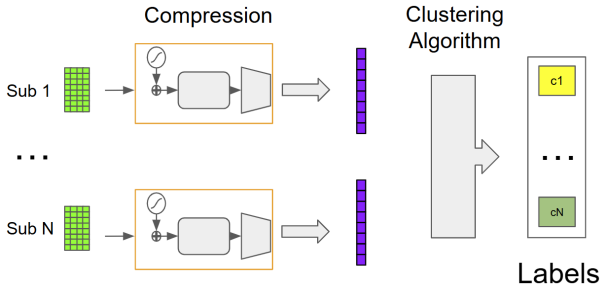

Figure 4: Clustering procedure.

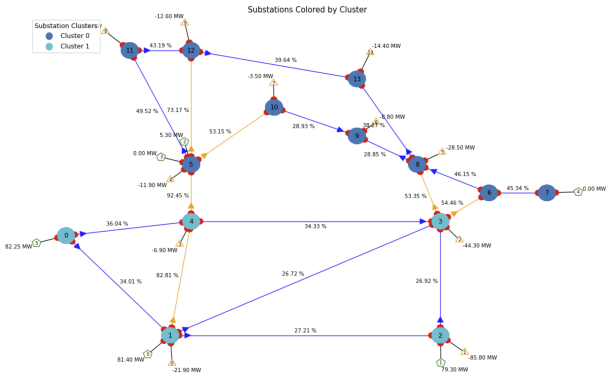

Figure 5: Most frequent clustering obtained by both Hierarchical clustering and DBSCAN.

6. Conclusions

This thesis described a framework for scalable real-time power grid control, by employing GNNs within a MARL architecture. The proposed algorithm was able to achieve a competitive performance despite decomposition of both action and observation spaces. Experimental results on the Grid2Op simulator confirmed the effectiveness of the approach, which consistently outperformed the Do-Nothing baseline and proved to be much more computationally efficient than the simulation-based Expert method. An ablation study further demonstrated the importance of each architectural component, particularly the GNN, the RL manager, and the DQfD module.

In spite of its strengths, the proposed approach still presents some limitations. The RL manager requires access to a complete global view of the environment, thus, preventing scalability. To overcome this limitation, multi-layer architectures represent a promising research direction. In these architectures, multiple high-level managers control different, independent portions of the power grid, identified through power grid decomposition methods.

Additionally, alternative action space reduction techniques should be explored to get rid of the simulation step performed at runtime, which inevitably affects the inference time. A promising direction is the N-1 criterion. This criterion retains only those configurations that ensure redundancy in line connections, where at least two lines connect a pair of buses. This redundancy ensures that if one line is disconnected, the other can still maintain the connection.

Finally, a dataset of expert demonstrations is not always available, especially in huge power grids in which expert strategies are unknown. Therefore, efforts to cope with the slow convergence observed in configurations without DQfD represents promising research directions. In this context, GNN-based transfer learning emerges as an interesting avenue to accelerate training. This involves using a GNN trained on a smaller power grid as initialization for a model operating on a larger grid.

References

[1] Jacob Adamczyk, Volodymyr Makarenko, Stas Tiomkin, and Rahul V. Kulkarni. Bootstrapped reward shaping, 2025.

[2] Matthijs de Jong, Jan Viebahn, and Yuliya Shapovalova. Generalizable graph neural networks for robust power grid topology control, 2025.

[3] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, Gabriel Dulac-Arnold, John Agapiou, Joel Z. Leibo, and Audrunas Gruslys. Deep q-learning from demonstrations. AAAI Press, 2018.

[4] Antoine Marot, Benjamin Donnot, Camilo Romero, Balthazar Donon, Marvin Lerousseau, Luca Veyrin-Forrer, and Isabelle Guyon. Learning to run a power network challenge for training topology controllers. *Electric Power Systems Research*, 2020.

[5] Erica van der Sar, Alessandro Zocca, and Sandjai Bhulai. Optimizing power grid topologies with reinforcement learning: A survey of methods and challenges, 2025.