A hybrid curriculum learning and tree search approach for network topology control

Meppelink, G.J. ; Rajaei, A.; Cremer, Jochen L.

DOI
10.1016/j.epsr.2025.111455

Publication date
2025

Document Version
Final published version

Published in
Electric Power Systems Research

Citation (APA)
Meppelink, G. J., Rajaei, A., & Cremer, J. L. (2025). A hybrid curriculum learning and tree search approach for network topology control. *Electric Power Systems Research*, *242*, Article 111455. https://doi.org/10.1016/j.epsr.2025.111455

Electric Power Systems Research 242 (2025) 111455

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

A hybrid curriculum learning and tree search approach for network topology control

G.J. Meppelink, A. Rajaei *, Jochen L. Cremer

Department of Electrical Sustainable Energy, Delft University of Technology, Delft, The Netherlands

ARTICLE INFO

Keywords:
Network topology control
Reinforcement learning
Curriculum learning
Monte Carlo tree search

ABSTRACT

Transmission network topology control offers cheap flexibility to system operators for mitigating grid congestion. However, finding the optimal sequence of topology actions remains a challenge due to the large number of possible actions. Although reinforcement learning (RL) approaches have attracted interest for long-term planning in large combinatorial action spaces, they encounter challenges such as training stability, sample efficiency, and unforeseen consequences of RL actions. Addressing these challenges, this paper proposes a hybrid curriculum-trained RL and Monte Carlo tree search (MCTS) approach to determine sequential topological actions for mitigating grid congestion. The curriculum-based approach stabilizes training by first pre-training a policy network through supervised imitation learning, followed by RL training. The policy network guides the MCTS to simulate promising future trajectories, mitigating unforeseen consequences and identifying long-term strategies to improve grid security. Moreover, the MCTS-verified actions are used for RL training, enhancing sample efficiency and training time. A distance factor is added to the MCTS, which improves convergence by prioritizing actions closer to congestion. Numerical results on the IEEE 118-bus system show that the proposed hybrid approach improves the timesteps survived by 30% compared to a standard RL approach, and by 5% compared to a brute-force baseline. Additionally, the inclusion of the distance factor increases the timesteps survived by 15%. These results highlight the potential of the proposed method for real-world applications of using sequential topological actions to effectively relieve grid congestion.

1. Introduction

Transmission grid congestion is a growing challenge, driven by rising electricity demand, aging infrastructure, and the integration of renewable energy sources (RES). Transmission network topology control has been highlighted as an under-exploited flexibility that can alleviate congestion by redirecting power flows by either line switching or substation reconfiguration. As an alternative to costly re-dispatch actions, topology control offers a more efficient way to relieve congestion, improve grid security, and reduce overall operational costs [1–3]. However, the combinatorial nature of the topology control problem poses significant challenges to identifying optimal topology control sequences in real time. As a result, system operators rely on their experience or predefined manuals, which can result in suboptimal grid performance or, in extreme cases, lead to system failures and blackouts [4]. Recent efforts, such as the "learning to run a power network" (L2RPN) competitions hosted by RTE, the French transmission system operator, have explored the use of artificial intelligence (AI), particularly reinforcement learning (RL), to determine sequential topology control actions [4–7]. These approaches can potentially identify action sequences that were not previously known by expert knowledge, offering new strategies for topology control. However, RL-based approaches face challenges in training stability, sample efficiency, and unforeseen consequences of RL actions. This paper addresses these challenges by proposing a novel approach based on curriculum learning (CL) and Monte Carlo tree search (MCTS) for sequential topology control.

Optimal transmission line switching (OTS) has been heavily investigated in the literature to reduce operational costs [8,9], reduce line flow and voltage violations [10,11], improve system reliability [12], and address the uncertainties of renewable energy sources [13]. Substation reconfiguration including busbar splitting/merging can also provide operational flexibility by rerouting the flows in the network [1, 14]. Fig. 1 depicts an example of congestion management by busbar splitting [4]. While optimal transmission line switching problem is computationally challenging for industrial power grids [9], the substation reconfiguration problem poses even more computational challenges due to the complex node-breaker modelling [15]. Authors in [15–18] propose mixed integer programming (MIP) formulations for the topology control problem. However, the approach in [15–18] does

* Corresponding author.
E-mail addresses: g.j.meppelink@student.tudelft.nl (G.J. Meppelink), a.rajaei@tudelft.nl (A. Rajaei), j.l.cremer@tudelft.nl (J.L. Cremer).

https://doi.org/10.1016/j.epsr.2025.111455
Received 23 July 2024; Received in revised form 25 October 2024; Accepted 20 January 2025
Available online 30 January 2025

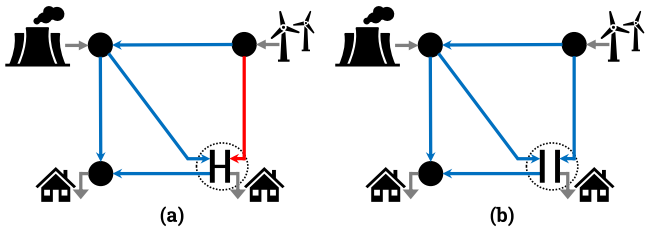

Fig. 1. Example of congestion management by substation reconfiguration. (a) busbar coupler is closed. (b) busbar splitting is applied by opening the busbar coupler.

not scale to industry-scale systems. [1] develops a heuristic approach based on sensitivity of line currents to the breaker position to find switching actions reducing congestion in the grid. [19] uses expert knowledge and heuristics to identify topological remedial actions to alleviate congestion. However, the developed approaches in [1,14–17,19] fail provide a sequence of control actions over a time horizon, limiting their effectiveness in dynamic grid environments [4].

Recent research has investigated AI-based approaches to overcome this limitation. In particular, RL algorithms are suited for problems involving a sequential decision making process, similar to the topology control problem [20]. Authors in [21] propose a deep duelling Q-network (DDQN) that is initialized with imitation learning. In [22], a Semi-Markov actor-critic algorithm is developed, which uses a graph neural network to extract graph-based information. [23] develops a planing algorithm that searches through the action set decided by a policy network. In [24], an RL agent based on proximal policy optimization (PPO) [25] and expert rules is proposed. Furthermore, similar expert heuristics, e.g. a reduced action space (RAS), are used by L2RPN competitors that underscore their practical importance [5,26]. However, these RL-based approaches [21–26] encounter challenges, including training instability, sample inefficiency, and unforeseen consequences of RL-proposed actions.

Curriculum-based RL for topology control, as proposed in [26,27], addresses the challenges of training stability and sample inefficiency by progressively exposing the RL agent to increasingly complex tasks, thereby improving learning efficiency and ensuring stable performance. However, CL does not account for the unforeseen consequences of actions, which can be drastic in practical grid operations. On the other hand, MCTS in [28,29] addresses unforeseen consequences by simulating future outcomes, guiding the agent towards long-term strategies. However, optimally training the policy and value networks requires a great amount of simulation trajectories. The proposed hybrid approach in this paper combines the strengths of CL and MCTS, offering advantages such as improved training stability, enhanced sample efficiency, identification of long-term strategies, and mitigation of unforeseen consequences.

This paper proposes a hybrid approach of curriculum-trained RL and MCTS to determine sequential topological actions for congestion management. The approach uses CL to stabilize the RL training process and improve its efficiency. Meanwhile, MCTS ensures that the agent's actions are guided by long-term strategies by simulating potential future outcomes. Initially, a policy network is pre-trained using supervised imitation learning on an offline dataset of expert optimal actions, providing a strong foundation for the agent. This policy network is then further refined through proximal policy optimization (PPO) RL training, allowing the agent to explore and learn from new actions. This stepwise CL training process addresses the common instability challenges of traditional RL methods. Furthermore, the policy network is integrated into an upper confidence bound (UCB) to guide the MCTS in identifying long-term strategies. By using MCTS-verified promising trajectories for RL training, the approach improves sample efficiency and accelerates the learning process. Additionally, a distance factor is added to the UCB to prioritize actions that are closer to congestion, accelerating MCTS convergence and facilitating scalability to large-scale power grids. Therefore, this hybrid approach identifies reliable long-term switching strategies to relieve congestion, improving grid security while reducing the reliance on costly redispatch actions, ultimately leading to lower operational costs.

The rest of the paper is organized as follows. Section 2 presents the proposed approach, outlined by the expert heuristics, the CL approach, the MCTS, and the distance factor. Section 3 presents the case studies on the IEEE 118-bus system. Section 4 concludes the paper.

2. Proposed hybrid curriculum learning and MCTS approach

This paper focuses on congestion management in transmission networks using sequential topological actions to prevent cascading failures that can lead to blackouts. The Grid2Op package [30], developed by RTE, formulates this problem as a Markov decision process (MDP), enabling realistic evaluations of sequential network operations [4–7]. Grid2Op uses Chronix2Grid [31] to generate synthetic but realistic consumption, renewable production, and economic dispatched productions chronics. The state includes a subset of the observation space, including nodal consumption and production, power line flows, and current topology. The action space includes continuous re-dispatching actions and discrete topological actions, with the latter offering a non-costly alternative to expensive re-dispatching.[1] However, the combinatorial nature of topological actions introduces significant computational challenges, which this paper addresses by focusing on optimal topological action selection.

2.1. Method overview

Fig. 2 depicts the proposed hybrid CL and MCTS approach during the testing phase. At each state s_t, first, a set of expert rules based on domain-knowledge and heuristics are checked to improve the performance and safety of the approach. Only unsafe states that are not resolved with the expert heuristics are passed to the proposed approach. Then, a MCTS process searches for the best possible sequence of actions $a \in \mathcal{A}$ that can lead to a safe state. The MCTS is guided by a CL-trained policy $\pi_\theta(a|s) : \mathcal{S} \rightarrow \mathcal{A}$. Finally, the action leading to the safest state is applied to the grid, proceeding to the next environment state. The proposed approach leverages the sample-efficient training approach of CL, while introducing an improved level of security by considering the simulation of promising actions through a guided MCTS that enhances the performance during training and testing. The search process is extended by including a distance factor to improve the convergence of MCTS.

2.2. Expert heuristics

Various expert heuristics are used throughout most approaches of the L2RPN competition, indicating their value. The similarities comprise of using a RAS, utilizing base-level rules, and a safety check [5,22–24,26–28]. These expert rules are based on domain knowledge and heuristics to limit negative effects of possible sub-optimal topology actions, and improve the performance of the approach.

2.2.1. Reduced action space

An RAS of the most useful actions is created using an offline study, allowing for vastly reduced computational time during training, while still enabling sufficient flexibility of topological actions for mitigating congestion. Extensive simulations are performed offline throughout varying scenarios, assessing actions that can be used for overflow reduction. From the original set of possible actions $a \in \mathcal{A}$, the RAS $a \in \mathcal{A}^R$ includes the top-N actions most used for flow reduction.

[1] We refer to [4–7,30] for complete description of the environment, including observation space, action space, data objects, and rewards.

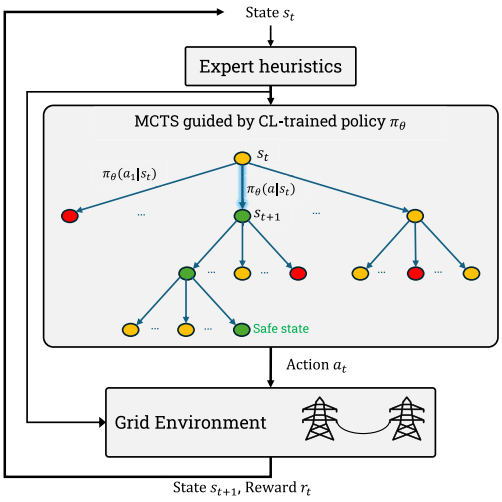

Fig. 2. The proposed hybrid approach during testing. The policy network π_θ, that is trained with curriculum learning, guides the MCTS to identify a long-term strategy to a safe state.

2.2.2. Base-level rules

A set of base-level rules are used to increase the resilience of the grid during normal (base) states. The first base-level rule ensures that all power lines are reconnected when possible. Secure grid performance is often contingent on keeping all power lines operational, as it increases the chances to offer flexibility by topological actions that can prevent congestion. The second base-level rule automates grid recovery to a default safe state after the congestion is alleviated. Multiple competitors in the L2RPN competition observed optimal grid resilience when all elements at the substations were connected to the same bus [5]. This default, fully-connected configuration maximizes connectivity and flow distribution, and provides maximum flexibility for the usage of topology control in future critical states. The third base-level rule subjects the agent to propose topology actions only when a set of system limits are violated. In particular, consider ρ as the maximum line loading of the grid for a given state. The agent only acts when the grid enters a state with $\rho > \rho^{max}$. The action threshold of ρ^{max} prevents agents from deteriorating grid security during safe states.

2.2.3. Safety check

Rather than executing the highest-ranking action suggested by the approach, a safety check step is introduced by simulating potential actions. This safety check ensures the selection of the safest sequence of actions. However, this approach selects actions based on predicted direct performance, which might not be the most optimal action in the long term. Note that the approach in this paper overcomes this limitation using the MCTS, which simulates possible future actions and thus takes the future outcomes into account.

2.3. Curriculum learning approach

In a CL approach, tasks are organized and presented in order of complexity to a learning algorithm. Neural networks (NNs) that have undergone CL can exhibit advanced responses, achieving improved generalization and quality of local minima [32] with fewer training samples. These advantages are essential in the context of transmission network topology control, where training convergence requires high computational complexity [27].

Fig. 3 presents the proposed CL approach. As Fig. 3-(a) shows, first, an offline dataset of state–action (s_t, a_t) pairs is generated, which indicates what action $a_t \in \mathcal{A}^R$ would an expert agent take in state $s_t \in \mathcal{S}$. To this end, in a overflowing state s_t, a brute-force approach simulates and assess all possible actions of the RAS to find the best action a_t^*. This state–action pair (s_t, a_t^*) is then saved in the offline dataset $\Omega^x = \{(s_t, a_t^*)\}$, indicating an expert policy that reduces overflows with topological actions.

In Fig. 3-(b), the *Junior* policy NN π_θ^J is defined as below:

$$\pi_\theta^J(a|s) : \mathcal{S} \rightarrow \mathcal{A}^R \tag{1}$$

where, θ is the weights of a standard feed forward NN. The Junior policy seeks to mimic the behaviour of the brute-force determined best actions in step (a). To this end, the Junior policy NN π_θ^J is trained using supervised imitation learning on the offline dataset generated in step (a):

$$\theta = \underset{\theta}{\text{argmin}} \quad \mathcal{L}_\theta^{SP}(\pi_\theta^J(.|s), a^*) \tag{2}$$

where $(s, a^*) \sim \Omega^x$ and $\mathcal{L}^{SP}$ is a supervised learning loss function, such as the cross entropy loss. Notably, the brute-force offline dataset only includes one-step actions, and not sequential actions. Therefore, the Junior policy may struggle to find optimal sequence of actions.

In Fig. 3-(c), the pre-trained Junior policy NN is improved upon through RL, resulting in the *Senior* policy NN π_θ^S. It is noteworthy that the policy network is used to guide the MCTS, which is discussed in the next section. The Senior NN is identical to the Junior NN, sharing its layer and neuron structure. The weights of the imitation learned Junior NN are used as starting point for the Senior NN. The Proximal policy optimization (PPO) [25] is used in this paper to train the Senior policy NN.

PPO is a policy gradient method that balances exploration and exploitation by optimizing a clipped surrogate loss function. Specifically, PPO minimizes a surrogate loss while preventing excessively large updates that could destabilize the policy. The surrogate loss is given by:

$$L_\theta^{PPO} = -\mathbb{E}_t \left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right] \tag{3}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio between the current policy and the old policy. $\hat{A}_t$ is the estimated advantage function at time step t, i.e., $\hat{A}_t = \hat{Q}(s_t, a_t) - \hat{V}(s_t)$. ϵ is a hyperparameter that controls the extent of clipping.

The clipping operation ensures that the policy update does not lead to large changes in the probability ratio $r_t(\theta)$. This helps stabilize the training process by preventing significant deviations that could result in performance degradation. The PPO algorithm alternates between sampling data from the environment and performing multiple epochs of stochastic gradient descent on the clipped surrogate loss $\mathcal{L}_\theta^{PPO}$. We refer to [25] for further information about PPO training.

In this way, the Senior policy NN π_θ^S refines the Junior policy, balancing between maintaining the quality of learned behaviour in step (b) and improving them further through RL training.

2.4. Monte Carlo tree search

The MCTS enhances the CL approach to overcome the limitations of brute-force and to enhance the reliability of topological actions, by considering potential future outcomes. Notably, MCTS-based agents such as [28,29] demonstrate exceptional success in environments with expansive action spaces. This success is due to the systematic exploration of different possible action paths, balancing the exploitation of promising actions with the exploration of gathering more information about uncertain ones. The MCTS search and simulate different sequences of actions depending on the expected value, which is based on a prior probability, as predicted by the policy network, and the visit count. This improves the probability of finding the (near) optimal sequence of actions, without having to exhaustively search all action trajectories.

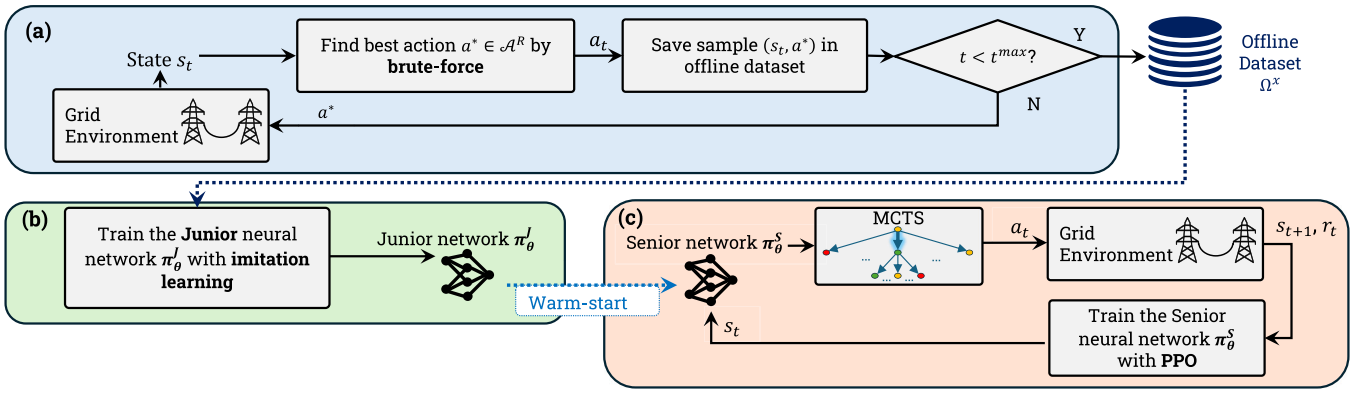

Fig. 3. The proposed training approach with curriculum learning. (a) Offline dataset of state–action pairs is generated using a brute-force method. (b) The junior network is trained with imitation learning on the offline dataset. (c) The senior network guides the MCTS and is trained by PPO-based reinforcement learning.

Each MCTS simulation includes a defined amount of iterations, traversing the search tree. The upper confidence bound (UCB) guides the action selection at each branching of the tree:

$$U(a|s) = \pi_\theta(a|s)\sqrt{\frac{\sum_{a'} N(s,a')}{1+N(s,a)}} \tag{4}$$

where $U(a|s)$ is the UCB of action a from state s, $\pi_\theta(a|s)$ is the prior probability of action a from state s, $N(s,a)$ is the visit count of action a from state s, and $\sum_{a'} N(s,a')$ is the total visit count for all actions $a' \in \mathcal{A}^R$ from state s. The UCB in (4) initially prioritize action selection with the highest probabilities based on the policy network. Subsequently, the UCB balances exploiting promising actions based on $\pi_\theta(a|s)$ with exploring less frequently tried actions based on visiting counts $N(s,a)$.

The MCTS simulation terminates after a defined number of iterations, or when the early stopping rule is triggered, e.g. if enough safe states have been found. The objective of the agent is to operate the grid for as long as possible. Therefore, after MCTS iterations, the state with maximum number of time steps (survived) and low value of ρ^{max} (maximum line loading) is selected, and the action leading there in the MCTS is applied. Fig. 4 shows an example of MCTS action selection. At the end of simulation, the visit counts of all traversed tree edges (i.e., $N(s,a)$) and the policy network weights are updated.

The MCTS-simulated sequences of actions leading to a safe state are further used to refine the policy network. In other words, the Junior policy network π^J_θ, which initially guides the MCTS, is refined through PPO training on the MCTS-generated trajectories, resulting in the Senior policy network π^S_θ. To this end, the wights θ are updated by multiple epochs of stochastic gradient descent on the PPO loss function $\mathcal{L}^{PPO}_\theta$ in (3) is applied.

Training with MCTS in the loop improves sample efficiency through MCTS simulations and considering only promising trajectories, instead of time consuming standard RL exploration methods. Moreover, the MCTS simulations provide an immediate feedback on actions without waiting for RL episode completion. This immediate feedback allows for the direct identification of better actions, significantly reducing the variance associated with standard RL approaches. As a result, the MCTS enhances the convergence to an effective policy and mitigates the risk of reaching a local optima. To maximize the efficiency of training, sample episodes are processed in parallel, with model parameters averaged after each iteration. The sample-efficient training approach efficiently uses computational resources for an accelerated overall learning process.

2.5. Distance guided search

The MCTS in the previous section adds a significant layer of security to the action selection process. However, if the initial training of the policy network is not optimal, the MCTS might struggle to rapidly identify viable solutions, hindering quick training. To address this issue, we propose a distance factor to guide the search tree. This modification streamlines the action selection process within the tree, enhancing the convergence rate of the training approach and ensuring more efficient learning.

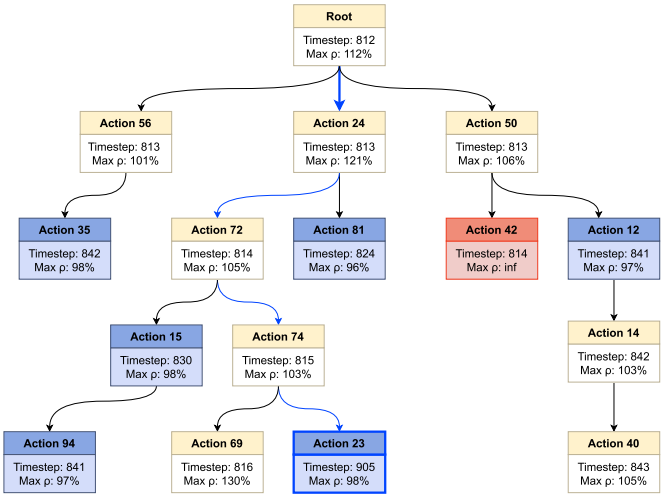

Fig. 4. MCTS action selection example. Nodes represent grid states, and root is the current state. Grey nodes are violating states, red nodes are black-out states and blue nodes are safe states. The safe grid state able to reach the maximum number of time steps (905) is selected, and the action leading there from the root is applied (action 24).

Topological interventions located closer to overflows tend to have a higher impact in reducing congestion [11,15,16]. As changing topologies in the MCTS would require exhaustive re-computation of metrics such as PTDFs, we use hops as an estimation of electrical distance, measured as the shortest path between an overflowing line and the location of an action. A fast breadth-first-search algorithm is used to determine the shortest path in hops. The distance factor is defined as:

$$DF(d) = \frac{1}{1+e^{(d-d^{Th})}} \tag{5}$$

where $DF(d)$ is the distance factor, d is the distance measured in hops, and d^{Th} is a hyper-parameter. The distance factor prioritize the actions with lower hop counts than d^{Th}. While this metric does not provide an accurate measure of the impact of an action on a congested line, it can offer a valuable insight on the probability of an action having an impact on the congested line.

In the context of the MCTS, the distance factor can tune the tree selection process by adjusting the UCB values at each state of the tree. This additive integration respects the prior action probabilities, while

guiding the decision-making process in favour of actions that are closer and, presumably, more impactful. The UCB with distance factor is:

$$U(a|s) = \pi_\theta(a|s) \cdot \sqrt{\frac{\sum_a N(s,a)}{1+N(s,a)}} \times (1 + DF(d)) \quad (6)$$

The distance factor serves as a guiding heuristic, initially steering the policy search, but gradually decreasing in influence as the training progresses. The scaling by the distance factor ensures that as the agent's policy improves and stabilizes, and the artificial bias from the distance factor diminishes, allowing the agent to rely on its learned strategy instead of the heuristic.

3. Case studies

3.1. Settings and test networks

All case studies are performed on the IEEE 118-bus system in the Grid2Op '*WCCI_L2RPN_2022*' environment [30]. 32 years of training data chronics with 5 min resolution is generated using Chronix2Grid package [31]. A test dataset consisting of 52 weekly scenarios with from the L2RPN 2022 competition is used, offering realistic variance in load, renewable generation and line outages [33]. The test episodes include an adversarial agent that removes lines to investigate the robustness of the proposed approach [6]. A separate validation dataset composed of 52 weekly episodes is used to determine the hyper-parameters of the models. For the test and the validation dataset, the adversarial agents are seeded randomly 5 different times, changing their attack locations and times, to reduce the outcome variance. While the adversarial agent simulates N-1 contingency cases, different grid structures are not considered. The imitation learning of the CL approach is done on 63,308 state-action pairs based on Section 2.3. The policy NN is trained in a custom Ray [34] environment, allowing the base-level rules to be integrated into the decision making process, while also assigning a positive reward over steps where no agent action was required. The gathered basic Grid2Op reward trains the RL agent [35]. All case studies are performed using DelftBlue's supercomputer Intel XEON E5-6248R 24C 3.0 GHz CPU cores [36]. Grid2Op 1.9.3, LightSim2Grid 0.7.3, and pandapower 2.11.1 package are used.

The performance of the proposed hybrid approach is compared with the following baselines:

- Do-Nothing: does not take any topological actions, but uses only the base-level rules of line re-connection and grid recovery.
- Brute-force: simulates through all actions in the RAS, and selects the action with the best predicted immediate flow reduction.
- Junior: the Junior policy network of the CL approach, that is trained with imitation learning. The top 25 predicted actions are simulated.
- Senior: the Senior policy network of the CL approach, that is initialized with the weights of the Junior NN and then trained with PPO. The top 25 predicted actions are simulated.
- The proposed hybrid MCTS and CL: uses the policy NN from the Junior and Senior for the prior predictions, to determine the MCTS trajectories. The iteration limit is set to 150. The MCTS-trained model takes the Junior NN for the priors, and has completed 3 full iterations of 1663 scenarios.
- The proposed MCTS-distance: utilizes the Junior NN for the prior predictions, including the distance bonus. The MCTS-distance trained model is trained for 3 full iterations of 1663 scenarios.

The Junior policy NN is based on the Binbinchen agent [37], where hyperparameter tuning is used to obtain layer and training parameters. An overview of these parameters can be seen in Table 1. The MCTS and MCTS-distance training use the Adam optimizer with the sparse categorical cross-entropy loss. Maximum timesteps survived is considered as a performance metric of the approach.

Table 1
Junior/Senior neural network parameters and values.

Parameter	Value
Input layer	1221 Variables
Hidden layer 1	400 Neurons
Dropout layer 1	0.25
Hidden layer 2	773 Neurons
Dropout layer 2	0.40
Hidden layer 3	1044 Neurons
Hidden layer 4	344 Neurons
Output layer linear	100 Actions
Activation	Relu activation
Batchsize	256
Initializer	Orthogonal
Learning rate	5e−5
Epochs	1000
Early stopping	100 steps

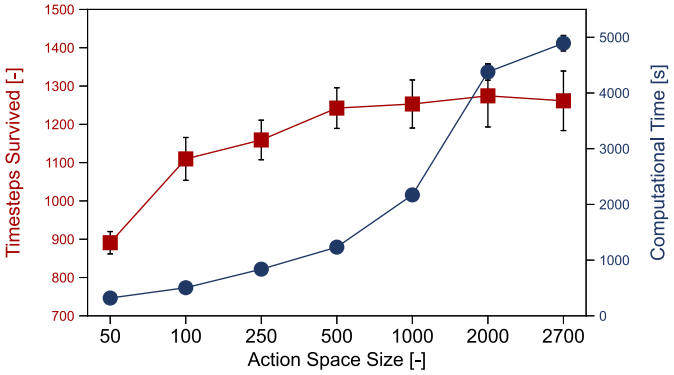

Fig. 5. Effect of the action space size, used by the brute-force algorithm, on the performance and computational time per scenario. Average values and standard deviations are indicated using markers and error bars, respectively.

The RAS only considers substation topological actions. Continuous, non-topological actions such as re-dispatching conventional generators, (dis-)charging of battery storage systems, and renewable curtailment are excluded to allow the evaluation of the proposed approach to determine substation topological actions. The size of the RAS is determined based on the performance of the brute-force agent considering various RAS sizes on the validation dataset. Fig. 5 show a clear trend between increased computational complexity and the usage of a larger RAS, while increase in performance stagnates after utilizing an RAS larger than 100. Due to computational power constraints, we use an RAS of 100 actions in this paper. Additionally, the maximum line loading threshold of $\rho^{max} = 98\%$ is assumed, i.e. the agent only acts when the grid has a maximum line loading above 98%. This threshold is determined by an offline study on various thresholds on the validation dataset.

3.2. CL and MCTS: Security and sample efficiency

Fig. 6 shows the timesteps survived per scenario of the test data set for the proposed hybrid CL and MCTS approach compared against the Do-Nothing, CL Junior and Senior, and Brute-force baselines. The Junior model shows marginal improvement compared to the Do-nothing agent, demonstrating the limitation of using supervised imitation learning alone. The Senior model shows a 35% improvement over the Junior model. This improvement is due to the further RL on MCTS trajectories that leads to longer-time strategies. Considering MCTS with the Junior and Senior models results in performance increases of 76% and 6%, respectively, with the Junior MCTS surpassing the Brute-force baseline by 5%. These results confirm the effectiveness of the MCTS in identifying action sequences that are more suitable for long-term operations, rather than focusing solely on immediate rewards. However,

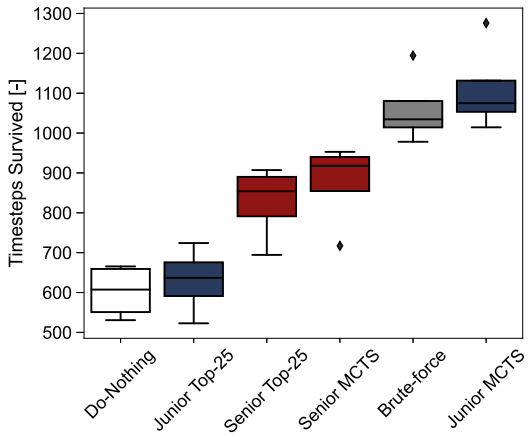

Fig. 6. Timesteps survived for the proposed hybrid MCTS and CL Junior and Senior models.

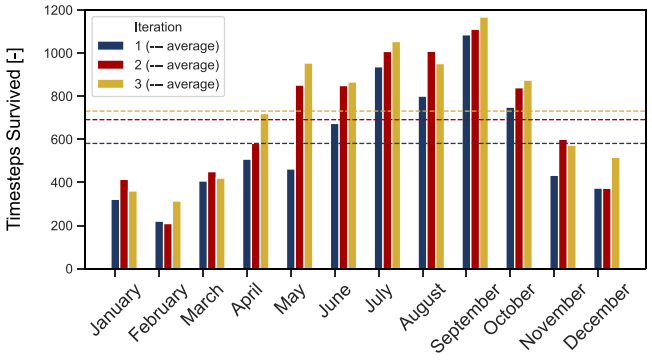

Fig. 7. Timesteps survived per iteration of the MCTS-training approach, split into monthly segments and parallel trained.

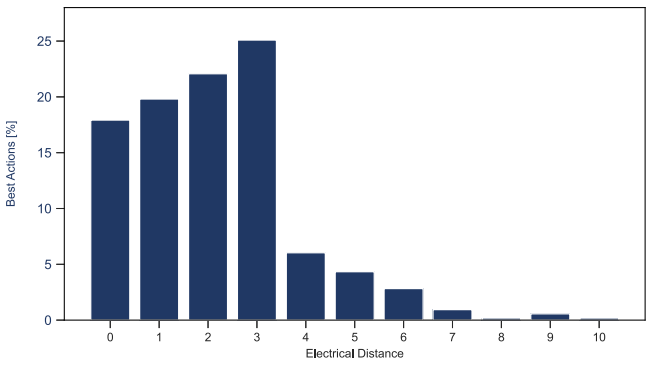

Fig. 8. Percentage of actions found that offer maximum flow reduction, and their distance to the overflow measured in hops.

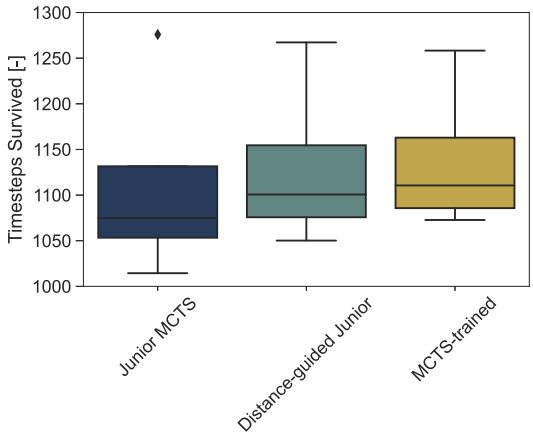

Fig. 9. Timesteps survived of the Junior MCTS, MCTS-trained, and the proposed distanced-guided MCTS for the test dataset.

the lower performance of the Senior MCTS compared to the Junior MCTS is due to the high confidence of the Junior NN in its prediction, compared to more diverse with lower confidence predictions of the Senior NN. This results in a shallower search by the Senior MCTS within the tree over 150 simulations. In other words, for a fixed number of simulations, the Junior MCTS searches deeper within the tree due to the less diverse predictions. While the consideration of CL and MCTS enhances performance, it does not inherently produce additive benefits, underscoring the importance of well-coordinated training strategies to maximize performance outcomes. This observation is further discussed in Section 3.4.

Fig. 7 shows the performance in timesteps survived per scenario of the training data set for 3 iterations of the MCTS-training approach. The results demonstrate the sample efficiency and stability of this training method, showing an overall improvement of more than 25% across the 3 iterations with only minor monthly dips throughout steadily improving performance. However, the stagnating performance increase showcases the effect of the adversarial agent, indicating a limit of what can be achieved using solely topological actions, especially when dealing with violations caused by unexpected outages of highly loaded lines.

3.3. Distance factor analysis

Fig. 8 shows the percentage of actions that achieve maximum flow reduction, along with their distance from the overflow, measured in hops. The top actions are identified using the brute-force baseline, i.e., simulating all $\mathcal{A}^R$ actions. As can be traced, most of the impactful actions are within 4 hops of the overflow. This supports our hypothesis that prioritizing lines closer to the congestion can achieve computational speed-up without compromising security gains. For the rest of the case studies, the $d^{Th} = 4$ in Eq. (5) is considered.

Fig. 9 shows the timesteps survived per scenario and Fig. 10 shows the MCTS iterations needed per step for the test data set, comparing the Junior MCTS, MCTS-trained and proposed distance-guided MCTS approach. The MCTS-trained model takes the Junior NN for the prior probabilities, and has completed 3 MCTS iterations of 1663 scenarios. The results indicate that the distance-guided approach achieves a similar performance increase to the MCTS-trained approach within 3 training iterations. Notably, this similar performance is achieved with approximately 15% fewer iterations per timestep.

Fig. 11 present the timesteps survived per scenario and Fig. 12 present the MCTS iterations needed per step for the training data set, over 3 iterations of the MCTS-trained and the proposed distance-guided MCTS-trained approach. The proposed distance-guided approach achieves improvements of 10% in iteration count and 15% in performance over the 3 iterations. This demonstrates the effectiveness of incorporating the distance factor, which helps in more rapidly identifying appropriate actions by prioritizing those closer to the congestion.

3.4. Discussion

The case studies present several advantages regarding the application of the CL, MCTS approach, and the proposed distance factor. The CL training stabilizes the RL training and lead to performance improvement of the Senior model compared to the Junior model. MCTS enhances performance by focusing on longer-time strategies and mitigating unforeseen consequences by simulations. By considering the

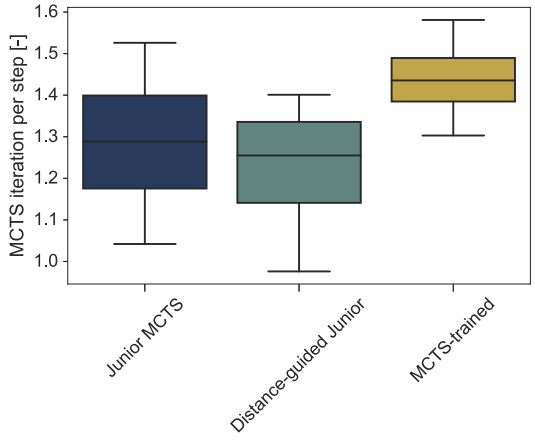

Fig. 10. Required MCTS iterations per timestep of the Junior MCTS, MCTS-trained and the proposed distanced-guided MCTS for the test dataset.

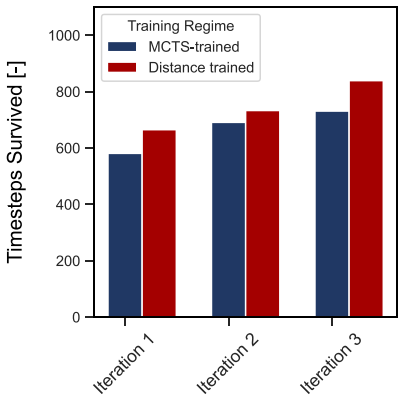

Fig. 11. Timesteps survived through 3 training iterations, for MCTS-trained and the proposed distance-trained approach.

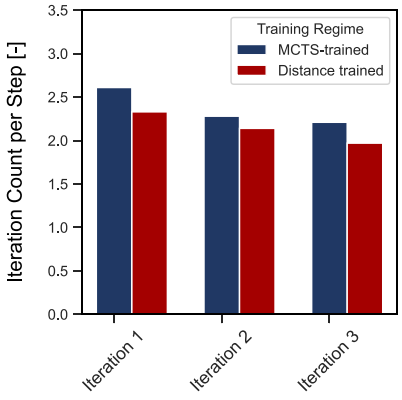

Fig. 12. Required MCTS iterations per timestep through 3 training iterations, for MCTS-trained and the proposed distance-trained approach.

MCTS into the training phase, the approach improves sample efficiency and performance, as simulation-tested actions are used during the policy training. The addition of a distance factor further accelerates convergence during MCTS simulations and training.

However, some limitation of the approach can be noted. Despite the inclusion of the distance factor, the MCTS remains computationally expensive. Although system operators can stop MCTS at any point and act on the best available solution, there is a trade-off between computation time and solution quality, which becomes critical in real-time applications. Additionally, the Senior MCTS did not outperform the Junior MCTS in the same amount of training iterations. The Junior model predicts fewer, more confident actions, whereas the Senior model predicts more diverse actions with less confidence due to the additional RL exploration. Therefore, the Senior model outperforms the Junior model when simulating top 25 predicted actions. However, when combined with MCTS, the Junior-MCTS model searches deeper within the tree due to its more focused predictions, while the Senior-MCTS explores more actions at each level, resulting in shallower tree exploration. That being said, the more diverse predictions of the Senior model can identify action sequences that were not previously known by an expert policy, enriching the experience of system operators.

In this study, the RAS of only substation topological actions were considered to evaluate how effectively our approach can identify these actions. However, this assumption creates a performance ceiling. Considering continuous, non-topological actions, such as re-dispatching conventional generators, charging battery storage systems, and renewable curtailment, can further enhance grid security. To this end, continuous actions can be considered through (1) new prediction heads in the policy network or (2) by optimizing continuous actions along promising topological trajectories using an optimal power flow (OPF) model within the MCTS approach.

While the primary objective in this work was maximizing the timesteps survived, real-world grid operations involve multiple objectives, including operational costs, long-term asset wear, and environmental impact [38]. Multi-objective RL approaches could be investigated to consider these factors and provide a set of Pareto optimal solutions for system operators to select from. Additionally, although the adversarial agent simulates N-1 contingencies to assess robustness, the proposed approach still requires retraining if the grid's base topology changes significantly.

4. Conclusion

This paper proposes a hybrid curriculum RL and MCTS approach for sequential topology control, aimed at mitigating grid congestion. The proposed approach uses CL to first pre-train a policy network through supervised imitation learning, followed by PPO. The policy network guides the MCTS to simulate potential action sequences, considering future outcomes for improved long-term performance. The MCTS is integrated into the policy training to increase training stability and sample efficiency. Additionally, a distance factor is introduced into the UCB to prioritize actions closer to congestion, improving convergence during MCTS simulation and training. Numerical case studies on the IEEE 118-bus system shows that the CL and MCTS stabilize learning and improve performance, with the proposed approach outperforming a brute-force baseline. Moreover, the proposed distance factor accelerates MCTS convergence, indicating potential improvements in scalability to larger grids by focusing on actions closer to congestion. However, MCTS remains computationally expensive, and there is a trade-off between computation time and solution quality, which is critical for real-time applications. Operators can stop MCTS at any point to act on the best available solution, providing flexibility in decision-making. Future work will focus on scalability to large-scale grids, considering continuous actions (e.g. re-dispatching) through the policy network or OPF in the MCTS, advanced electrical distance methods with the bus split distribution factors [39], adaptability to different grids, and multi-objective RL approaches to consider different operational objectives, such as security, cost, and asset wear.

CRediT authorship contribution statement

G.J. Meppelink: Writing – original draft, Software, Methodology, Investigation. **A. Rajaei:** Writing – review & editing, Supervision, Conceptualization. **Jochen L. Cremer:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors are thankful to J. Viebahn from TenneT, A. Marot, and B. Donnot from Reseau de Transport d' Electricite, and N. Lair from Artelys who provided valuable discussions.

Data availability

Data will be made available on request.

References

[1] S. Babaeinejadsarookolaee, B. Park, B. Lesieutre, C.L. DeMarco, Transmission congestion management via node-breaker topology control, IEEE Syst. J. (2023).
[2] M. Numan, M.F. Abbas, M. Yousif, S.S. Ghoneim, A. Mohammad, A. Noorwali, The role of optimal transmission switching in enhancing grid flexibility: A review, IEEE Access (2023).
[3] H. Bawayan, M. Younis, Mitigating failure propagation in microgrids through topology reconfiguration, Sustain. Energy Grids Netw. 23 (2020) 100363.
[4] A. Marot, B. Donnot, C. Romero, B. Donon, M. Lerousseau, L. Veyrin-Forrer, I. Guyon, Learning to run a power network challenge for training topology controllers, Electr. Power Syst. Res. 189 (2020) 106635.
[5] A. Marot, B. Donnot, G. Dulac-Arnold, A. Kelly, A. O'Sullivan, J. Viebahn, M. Awad, I. Guyon, P. Panciatici, C. Romero, Learning to run a power network challenge: A retrospective analysis, in: NeurIPS 2020 Competition and Demonstration Track, PMLR, 2021, pp. 112–132.
[6] A. Marot, et al., L2rpn: Learning to Run a Power Network in a Sustainable World Neurips2020 Challenge Design, White Paper, Réseau de Transport d'Électricité, Paris, France, 2020.
[7] A. Marot, B. Donnot, K. Chaouache, A. Kelly, Q. Huang, R.-R. Hossain, J.L. Cremer, Learning to run a power network with trust, Electr. Power Syst. Res. 212 (2022) 108487.
[8] E.B. Fisher, R.P. O'Neill, M.C. Ferris, Optimal transmission switching, IEEE Trans. Power Syst. 23 (3) (2008) 1346–1355.
[9] C. Crozier, K. Baker, B. Toomey, Feasible region-based heuristics for optimal transmission switching, Sustain. Energy Grids Netw. 30 (2022) 100628.
[10] M. Khanabadi, H. Ghasemi, M. Doostizadeh, Optimal transmission switching considering voltage security and N-1 contingency analysis, IEEE Trans. Power Syst. 28 (1) (2012) 542–550.
[11] X. Li, P. Balasubramanian, M. Sahraei-Ardakani, M. Abdi-Khorsand, K.W. Hedman, R. Podmore, Real-time contingency analysis with corrective transmission switching, IEEE Trans. Power Syst. 32 (4) (2016) 2604–2617.
[12] X. Li, P. Balasubramanian, M. Abdi-Khorsand, A.S. Korad, K.W. Hedman, Effect of topology control on system reliability: TVA test case, in: CIGRE US National Committee Grid of the Future Symposium, 2014.
[13] J. Aghaei, A. Nikoobakht, M. Mardaneh, M. Shafie-khah, J.P. Catalão, Transmission switching, demand response and energy storage systems in an innovative integrated scheme for managing the uncertainty of wind power generation, Int. J. Electr. Power Energy Syst. 98 (2018) 72–84.
[14] L. Wang, H.-D. Chiang, Bus-bar splitting for enhancing voltage stability under contingencies, Sustain. Energy Grids Netw. 34 (2023) 101010.
[15] M. Heidarifar, P. Andrianesis, P. Ruiz, M.C. Caramanis, I.C. Paschalidis, An optimal transmission line switching and bus splitting heuristic incorporating AC and N-1 contingency constraints, Int. J. Electr. Power Energy Syst. 133 (2021) 107278.
[16] M. Heidarifar, H. Ghasemi, A network topology optimization model based on substation and node-breaker modeling, IEEE Trans. Power Syst. 31 (1) (2015) 247–255.
[17] E.A. Goldis, P.A. Ruiz, M.C. Caramanis, X. Li, C.R. Philbrick, A.M. Rudkevich, Shift factor-based SCOPF topology control MIP formulations with substation configurations, IEEE Trans. Power Syst. 32 (2) (2016) 1179–1190.
[18] A. Ewerszumrode, N. Erle, S. Krahl, A. Moser, An iterative approach to grid topology and redispatch optimization in congestion management, Electr. Power Syst. Res. 234 (2024) 110700.
[19] A. Marot, B. Donnot, S. Tazi, P. Panciatici, Expert system for topological remedial action discovery in smart grids, in: Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion, MEDPOWER 2018, IET, 2018, pp. 1–6.
[20] X. Chen, G. Qu, Y. Tang, S. Low, N. Li, Reinforcement learning for selective key applications in power systems: Recent advances and future challenges, IEEE Trans. Smart Grid 13 (4) (2022) 2935–2958.
[21] T. Lan, J. Duan, B. Zhang, D. Shi, Z. Wang, R. Diao, X. Zhang, AI-based autonomous line flow control via topology adjustment for maximizing time-series ATCs, in: 2020 IEEE Power & Energy Society General Meeting, PESGM, IEEE, 2020, pp. 1–5.
[22] D. Yoon, S. Hong, B.-J. Lee, K.-E. Kim, Winning the l2rpn challenge: Power grid management via semi-markov afterstate actor-critic, in: International Conference on Learning Representations, 2020.
[23] B. Zhou, H. Zeng, Y. Liu, K. Li, F. Wang, H. Tian, Action set based policy optimization for safe power grid management, in: Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part V 21, Springer, 2021, pp. 168–181.
[24] A. Chauhan, M. Baranwal, A. Basumatary, Powrl: A reinforcement learning framework for robust management of power networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, 2023, pp. 14757–14764, 12.
[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, 2017, arXiv preprint arXiv:1707.06347.
[26] M. Lehna, J. Viebahn, A. Marot, S. Tomforde, C. Scholz, Managing power grids through topology actions: A comparative study between advanced rule-based and reinforcement learning agents, Energy AI 14 (2023) 100276.
[27] A.R.R. Matavalam, K.P. Guddanti, Y. Weng, V. Ajjarapu, Curriculum based reinforcement learning of grid topology controllers to prevent thermal cascading, IEEE Trans. Power Syst. (2022).
[28] M. Dorfer, A.R. Fuxjäger, K. Kozak, P.M. Blies, M. Wasserer, Power grid congestion management via topology optimization with AlphaZero, 2022, arXiv preprint arXiv:2211.05612.
[29] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al., A general reinforcement learning algorithm that masters chess, shogi, and go through self-play, Science 362 (6419) (2018) 1140–1144.
[30] B. Donnot, Grid2op- a testbed platform to model sequential decision making in power systems, 2020, [Online]. Available: https://GitHub.com/Grid2Op/grid2op.
[31] B. Donnot, ChroniX2Grid - The extensive PowerGrid time-serie generator, [Online]. Available: https://github.com/Grid2op/chronix2grid.
[32] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 41–48.
[33] G. Serré, E. Boguslawski, B. Donnot, A. Pavão, I. Guyon, A. Marot, Reinforcement learning for energies of the future and carbon neutrality: A challenge design, in: SSCI 2022-IEEE Symposium Series on Computational Intelligence, 2022.
[34] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M.I. Jordan, et al., Ray: A distributed framework for emerging {ai} applications, in: 13th USENIX Symposium on Operating Systems Design and Implementation, OSDI 18, 2018, pp. 561–577.
[35] RTÉ France, Grid2op documentation, 2019, Revision a819a777. [Online]. Available: https://grid2op.readthedocs.io/en/latest/.
[36] Delft High Performance Computing Centre (DHPC), DelftBlue supercomputer (phase 1), 2022, https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1.
[37] EI innovation lab, huawei cloud, huawei technologies. NeurIPS competition 2020: Learning to run a power network (L2RPN) - robustness track. 2, 2020, [Online]. Available: https://github.com/AsprinChina/L2RPN_NIPS_2020_a_PPO_Solution.
[38] J. Viebahn, M. Naglic, A. Marot, B. Donnot, S.H. Tindemans, Potential and challenges of AI-powered decision support for short-term system operations, in: CIGRE Session 2022, 2022.
[39] J. van Dijk, J. Viebahn, B. Cijsouw, J. van Casteren, Bus split distribution factors, IEEE Trans. Power Syst. (2023).