A Reinforcement Learning Framework for Optimal Operation and Maintenance of Power Grids

R. Rocchetta[a], L. Bellani[b], M. Compare[b,c], E Zio[b,c,d,e], E Patelli*[a]

[a] Institute for Risk and Uncertainty, Liverpool University, United Kingdom
[b] Aramis s.r.l., Milano, Italy
[c] Energy Department, Politecnico di Milano,Italy
[d] MINES ParisTech, PSL Research University, CRC, Sophia Antipolis, France
[e] Eminent Scholar, Department of Nuclear Engineering, College of Engineering, Kyung Hee University, Republic of Korea

Abstract

We develop a Reinforcement Learning framework for the optimal management of the operation and maintenance of power grids equipped with prognostics and health management capabilities. Reinforcement learning exploits the information about the health state of the grid components. Optimal actions are identified maximizing the expected profit, considering the aleatory uncertainties in the environment. To extend the applicability of the proposed approach to realistic problems with large and continuous state spaces, we use Artificial Neural Networks (ANN) tools to replace the tabular representation of the state-action value function. The non-tabular Reinforcement Learning algorithm adopting an ANN ensemble is designed and tested on the scaled-down power grid case study, which includes renewable energy sources, controllable generators, maintenance delays and prognostics and health management devices. The method strengths and weaknesses are identified by comparison to the reference Bellman's optimally. Results show good approximation capability of Q-learning with ANN, and that the proposed framework outperforms expert-based solutions to grid operation and maintenance management.

Keywords: Reinforcement Learning, Artificial Neural Networks, Prognostic and Health Management, Operation and Maintenance, Power Grid,

*Corresponding author: edoardo.patelli@liverpool.ac.uk

Uncertainty

1. Introduction

Power Grids are critical infrastructures designed to satisfy the electric power needs of industrial and residential customers. Power Grids are complex systems including many components and subsystems, which are intertwined to each other and affected by degradation and aging due to a variety of processes (e.g. creepage discharge [1], loading and unloading cycles [2], weather-induced fatigue [3], etc.). Maximizing the Power Grid profitability by the a safe and reliable delivery of power is of primary importance for grid operators. This requires developing sound decision-making frameworks, which account for both the complexity of the asset and the uncertainties on its operational conditions, components degradation processes, failure behaviors, external environment, etc.

Nowadays, Power Grid Operation and Maintenance (O&M) management is enhanced by the possibility of equipping the Power Grid components with Prognostics and Health Management (PHM) capabilities, for tracking and managing the evolution of their health states so as to maintain their functionality [4]. This capability can be exploited by Power Grid operators to further increase the profitability of their assets, e.g. with a smarter control of road lights [5]-[6], exploiting wide are control of wind farms [7] or with a better microgrid control [8] and management [9]. However, embedding PHM in the existing Power Grid O&M policies requires addressing a number of challenges [10]. In this paper, we present a framework based on Reinforcement Learning [11]-[12], for settings the generator power outputs and the schedule of preventive maintenance actions in a way to maximize the Power Grid load balance and the expected profit over an infinite time horizon, while considering the uncertainty of power production from Renewable Energy Sources, power loads and components failure behaviors. Reinforcement Learning has been used to solve a variety of realistic control and decision-making issues in the presence of uncertainty, but with a few applications to Power Grid management. For instance, Reinforcement Learning has been applied to address the generators load frequency control problem [13], the unit commitment problem [14], to enhance the power system transient stability [15] and to address customers' private preferences in the electricity market [16]. Furthermore, the economic dispatch [17] and the auction based pricing issues [18] have also been tackled using Reinforcement Learning. In [19], a Q-learning approach has been proposed to solve constrained load flow and reactive power control problems in Power Grids. In [9], a Reinforcement Learning-based optimization scheme has been designed for microgrid consumers actions management, and accounting for renewable volatility and environmental uncertainty. In [20], a comparison between Reinforcement Learning and a predictive control model has been presented for a Power Grid damping problem. In [21] a review of the application of reinforcement learning for demand response is proposed, whereas in [8], the authors have reviewed recent advancements in intelligent control of microgrids, which include a few Reinforcement Learning methods. However, none of the revised works employs Reinforcement Learning to find optimal O&M policies for Power Grids with degrading elements and equipped with PHM capabilities. Moreover, these works mainly apply basic Reinforcement Learning algorithms (e.g., the SARSA(λ) and Q-learning methods [12]), which rely on a memory intensive tabular representation of the state-action value function Q. The main drawback of these tabular methods lies in their limited applicability to realistic, large-scale problems, characterized by highly-dimensional state-action spaces. In those situations, the memory usage becomes burdensome and the computational times are intractable. To extend the applicability of Reinforcement Learning methods to problems with arbitrarily large state spaces, regression tools can be adopted to replace the tabular representation of Q (refer to [12] for a general overview on algorithms for RL and [22] for a introduction to deep RL).

In [23], a deep Q-learning strategy for optimal energy management of hybrid electric buses is proposed. In [24], Reinforcement Learning method is used to find the optimal incentive rates for a demand-response problem for smart grids. Real-time performance was augmented with the aid of deep neural networks.

Two RL techniques based on Deep Q-learning and Gibbs deep policy gradient are applied to physical models for smart grids in [25]. In [26], a RL method for dynamic load shedding is investigated for short-term voltage control; the southern China Power Grid model is used as a test system. In [27], RL for residential demand response control is investigated. However, only tabular Q-learning methods are investigated. To the best authors knowledge, none of the reviewed work proposed a non-tabular solution to operational and maintenance scheduling of power grid equipped with PHM devices.

In this paper, to extend the applicability of the proposed Reinforcement Learning method, we use Artificial Neural Networks (ANNs), due to their approximation power and good scalability propriety. The resulting Reinforcement Learning algorithm enables tackling highly-dimensional optimization problems and its effectiveness is investigated on a scaled-down test system. This example allows showing that Reinforcement Learning can really exploit the information provided by PHM to increase the Power Grid profitability.

The rest of this work is organized as follows: Section 2 presents the Reinforcement Learning framework for optimal O&M of Power Grids in the presence of uncertainty; a scaled-down power grid application is proposed in Section 3, whereas the results and limitations of Reinforcement Learning for Power Grid O&M are discussed in Sections 4; Section 6 closes the paper.

2. Modeling framework for optimal decision making under uncertainty

In the Reinforcement Learning paradigm, an agent (i.e. the controller and decision maker) learns from the interaction with the environment (e.g. the grid) by observing states, collecting gains and losses (i.e. rewards) and selecting actions to maximize the future revenues, considering the aleatory uncertainties in the environment behavior. On-line Reinforcement Learning methods can tackle realistic control problems through direct interaction with the environment. However, off-line (model-based) Reinforcement Learning methods are generally adopted for safety-critical systems such as power grids [28], due to the unacceptable risks associated with exploratory actions [28].

Developing an off-line Reinforcement Learning framework for Power Grid O&M management requires defining the environment and its stochastic behavior, the actions that the agent can take in every state of the environment and their effects on the grid and the reward generated. These are formalized below.

2.1. Environment State

Consider a Power Grid made up of elements $C = \{1, ..., N\}$, physically and/or functionally interconnected, according to the given grid structure. Similarly to [10], the features of the grid elements defining the environment are the n^d degradation mechanisms affecting the degrading components $d \in D \subseteq C$ and the n^p setting variables of power sources $p \in P \subseteq C$. For simplicity, we assume $D = \{1, ..., |D|\}$, $P = \{|D| + 1, ..., |D| + |P|\}$ and $|D| + |P| \leq N$. The extension of the model to more complex settings can be found in [10].

Every degradation mechanism evolves independently from the others, obeying a Markov process that models the stochastic transitions from state $s_i^d(t)$ at time t to the next state $s_i^d(t+1)$, where $s_i^d(t) \in \{1, ..., S_i^d\}$, $\forall t$, $d \in D, i = 1, ..., n^d$. These degradation states are estimated by the PHM systems (e.g., [29]).

Similarly, a Markov process defines the stochastic transitions of the p-th power setting variable from $s_j^p(t)$ at time t to the next state $s_j^p(t+1)$, where $s_j^p(t) \in \{1, ..., S_j^p\}$, $\forall t$, $p \in P, j = 1, ..., n^p$. Generally, these transitions depend on exogenous factors such as the weather conditions.

Then, system state vector $\mathbf{S} \in \mathcal{S}$ at time t reads:

$$\mathbf{S}_t = \left[\; s_1^1(t), s_2^1(t), \ldots, s_{n^{|P|+|D|}}^{|P|+|D|}(t) \; \right] \in \mathcal{S} \tag{1}$$

where $\mathcal{S} = \bigtimes_{\substack{f=1,\ldots,n^c \\ c=1,\ldots,|P|+|D|}} \{1, ..., S_f^c\}$.

2.2. Actions

Actions can be performed on the grid components $g \in G \subseteq C$ at each t. The system action vector $\mathbf{a} \in \mathcal{A}$ at time t is:

$$\mathbf{a}_t = \left[\; a_{g_1}(t), \ldots, a_{g_e}(t), \ldots, a_{|g|_{|G|}}(t) \; \right] \in \mathcal{A} \tag{2}$$

where action a_{g_ϱ} is selected for component $g_\varrho \in G$ among a set of mutually exclusive actions $a_{g_\varrho} \in A_{g_\varrho}$, $\varrho = 1, ..., |G|$, $\mathcal{A} = \mathop{\times}_{\varrho=1,...|G|} A_{g_\rho}$. The action set A_{g_ϱ} includes both operational actions (e.g. closure of a valve, generator power ramp up, etc.) and maintenance actions. Specifically, Corrective Maintenance (CM) and Preventive Maintenance (PM) are the maintenance actions considered in this paper. When CM action is performed to fix a faulty component, which is put from an out-of-service condition to a in-service, As-Good-As-New (AGAN) condition. Differently, predictive maintenance can be performed on an in-service, non-faulty (but degraded), component, to improve its degradation state.

Constraints can be defined for reducing A_{g_ϱ} to a subset $\hat{A}_{g_\varrho}(\mathbf{S}) \subseteq A_{g_\varrho}$, to take into account that some actions are not allowed in particular states. For example, Corrective Maintenance (CM), cannot be taken on As-Good-As-New (AGAN) components and, similarly, it is the only possible action for failed components. In an opportunistic view [10], both Preventive Maintenance (PM) and CM actions are assumed to restore the AGAN state for each component. An example of Markov process for a 4 degradation state component is presented in Figure 1.

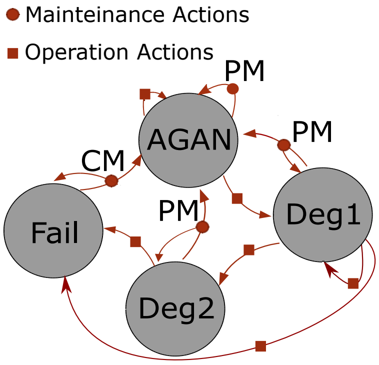

Figure 1: The Markov Decision Process associated to the health state of a degrading component; circle markers indicate maintenance actions whereas squared markers indicate operational actions.

2.3. Stochastic behavior of the environment state

As mentioned before, the development of a Reinforcement Learning framework for optimal O&M of Power Grids has to necessarily rely on a model of the stochastic behavior of the environment. We assume that this is completely defined by transition probability matrices associated to each feature $f = 1, ..., n^c$ of each component $c = 1, ..., |P| + |D|$ and to each action $\mathbf{a} \in \mathcal{A}$:

$$\mathcal{P}^{\mathbf{a}}_{c,f} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,S^c_f} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,S^c_f} \\ \vdots & \vdots & \ddots & \vdots \\ p_{S^c_f,1} & p_{S^c_f,2} & \cdots & p_{S^c_f,S^c_f} \end{bmatrix}^{\mathbf{a}}_{c,f} \tag{3}$$

where $p_{i,j}$ represents the probability $\mathcal{P}^{\mathbf{a}}_{c,f}(s_j|\mathbf{a}, s_i)$ of having a transition of component c from state i to state j of feature f, conditional to the action $\mathbf{a}$, $\sum_{j=1}^{n^c} p_{i,j} = 1$.

This matrix-based representation of the environment behavior is not mandatory

to develop a Reinforcement Learning framework. However, it allows applying dynamic programming algorithms that can provide the Bellman's optimal O&M policy with a pre-fixed, arbitrarily small error ([11]). This reference true solution is necessary to meet the objective of this study, which is the investigation of the benefits achievable from the application of Reinforcement Learning methods to optimal Power Grid O&M, provided that these methods must not be tabular for their application to realistic Power Grid settings.
The algorithm used to find the reference solution is reported in Appendix 6.

2.4. Rewards

Rewards are case-specific and obtained by developing a cost-benefit model, which evaluates how good the transition from one state to another is, given that $\mathbf{a}$ is taken:

$$R_t = R\left(\mathbf{S}_t \ , \ \mathbf{a}_t \ , \ \mathbf{S}_{t+1}\right) \ \in \mathbb{R}$$

Generally speaking, there are no restriction on the definition of a reward function. However, a well-suited reward function will indeed help the agent converging faster to an optimal solution [30]. Further specifications will depend strongly on the specific RL problem at hand and, thus, will be provided in section 3.3.

2.5. A non-tabular Reinforcement Learning algorithm

Generally speaking, the goal of Reinforcement Learning for strategy optimization is to maximize the action-value function $Q_{\pi^*}(\mathbf{S}, \mathbf{a})$, which provides an estimation of cumulated discounted future revenues when action $\mathbf{a}$ is taken in state $\mathbf{S}$, following the optimal policy π^*:

$$Q_{\pi^*}(\mathbf{S}, \mathbf{a}) = \mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty} \gamma^t R(t) | \mathbf{S}, \mathbf{a}\right] \tag{4}$$

We develop a Reinforcement Learning algorithm which uses an ensemble of ANNs to interpolate between state-action pairs, which helps to reduce the number of episodes needed to approximate the Q function.
Figure 2 graphically displays an episode run within the algorithm. In details, we estimate the value of $Q_\pi(\mathbf{S}_t, \mathbf{a}_t)$ using a different ANN for each action, with

network weights $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{|\mathcal{A}|}$, respectively. Network $\mathcal{N}_l$, $l = 1, ...|\mathcal{A}|$, receives in input the state vector $\mathbf{S}_t$ and returns the approximated value $\hat{q}_l(\mathbf{S}_t|\boldsymbol{\mu}_l)$ of $Q_\pi(\mathbf{S}_t, \mathbf{a}_t = a_l)$.

To speed up the training of the ANNs ([31]), we initially apply a standard supervised training over a batch of relatively large size n_{ei}, to set weights $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{|\mathcal{A}|}$. To collect this batch, we randomly sample the first state $\mathbf{S}_1$ and, then, move $n_{ei} + \Phi$ steps forward by uniformly sampling from the set of applicable actions and collecting the transitions $\mathbf{S}_t, \mathbf{a}_t \rightarrow \mathbf{S}_{t+1}, \mathbf{a}_{t+1}$ with the corresponding rewards $R_t, t = 1, ..., n_{ei} + \Phi - 1$. These transitions are provided by a model of the grid behavior.

Every network $\mathcal{N}_l, l \in \{1, \ldots, |\mathcal{A}|\}$, is trained on the set of states $\{\mathbf{S}_t | t = 1, ..., n_{ei}, \mathbf{a}_t = l\}$ in which the l-th action is taken, whereas the reward that the ANN learns is the Monte-Carlo estimate Y_t of $Q_\pi(\mathbf{S}_t, \mathbf{a}_t)$:

$$Y_t = \sum_{t'=t}^{t+\Phi} \gamma^{t'-t} \cdot R_{t'} \tag{5}$$

After this initial training, we apply Q-learning (e.g., [30],[12]) to find the ANN approximation of the optimal $Q_{\pi^*}(\mathbf{S}_t, \mathbf{a}_t)$. Namely, every time the state $\mathbf{S}_t$ is visited, the action $\mathbf{a}_t$ is selected among all available actions according to the $\epsilon-$greedy policy π: the learning agent selects exploitative actions (i.e., the action with the largest value, maximizing the expected future rewards) with probability $1 - \epsilon$, or exploratory actions, randomly sampled from the other feasible actions, with probability ϵ.

The immediate reward and the next state is observed, and weights $\boldsymbol{\mu}_{\mathbf{a}_t}$ of network $\mathcal{N}_{\mathbf{a}_t}$ are updated: a single run of the back-propagation algorithm is done ([32],[33]) using $R_t + \gamma \cdot \max_{l \in \{1,\ldots,|\mathcal{A}|\}} \hat{q}_l(\mathbf{S}_{t+1}|\boldsymbol{\mu}_l)$ as target value (Equation 6). This yields the following updating:

$$\boldsymbol{\mu}_{\mathbf{a}_t} \leftarrow \boldsymbol{\mu}_{\mathbf{a}_t} + \alpha_{\mathbf{a}_t} \cdot [R_t + \gamma \cdot \max_{l \in \{1,\ldots,|\mathcal{A}|\}} \hat{q}_l(\mathbf{S}_{t+1}|\boldsymbol{\mu}_l) - \hat{q}_{\mathbf{a}_t}(\mathbf{S}_t|\boldsymbol{\mu}_{\mathbf{a}_t})] \cdot \nabla \hat{q}_{\mathbf{a}_t}(\mathbf{S}_t|\boldsymbol{\mu}_{\mathbf{a}_t}) \tag{6}$$

where $\alpha_{\mathbf{a}_t} > 0$ is the value of the learning rate associated to $\mathcal{N}_{\mathbf{a}_t}$ ([30]).

Notice that the accuracy of the estimates provided by the proposed algorithm strongly depends on the frequency at which the actions are taken in every state:

the larger the frequency, the larger the information from which the network can learn the state-action value [30]. In real industrial applications, where systems spend most of the time in states of normal operation ([34]), this may entail a bias or large variance in the ANN estimations of $Q_\pi(\mathbf{S}_t, \mathbf{a}_t)$ for rarely visited states. To overcome this issue, we increase the exploration by dividing the simulation of the system, and its interactions with the environment and O&M decisions, into episodes of fixed length T. Thus, we run N_{ei} episodes, each one entailing T decisions; at the beginning of each episode, we sample the first state uniformly over all states. This procedure increases the frequency of visits to highly degraded states and reduces the estimation error. At each episode $ei \in \{1, \ldots, N_{ei}\}$, we decrease the exploration rate $\epsilon = \epsilon_{ei}$ according to $\epsilon = \epsilon_0 \cdot \tau_\epsilon^{ei}$, and the learning rate $\alpha_l = \alpha_0 \cdot (\frac{1}{1+K_\alpha \cdot t_l})$, where α_0 is the initial value, K_α is the decay coefficient and t_l counts the number of times the network $\mathcal{N}_l$ has been trained ([30]).

3. Case study

A scaled-down Power Grid case study is considered to apply the Reinforcement Learning decision making framework. The Power Grid includes: 2 controllable generators, 5 cables for power transmission, and 2 renewable energy sources which provide electric power to 2 connected loads depending on the (random) weather conditions (Figure 3). Then, $|C|$=11. The two traditional generators are operated to minimize power unbalances on the grid (Figure 3). We assume that these generators, and links 3 and 4, are affected by degradation and are equipped with PHM capabilities to inform the decision-maker on their degradation states. Then, $D = \{1, 2, 3, 4\}$. The two loads and the two renewable generators define the grid power setting, $P = \{5, 6, 7, 8\}$

3.1. States and actions

We consider $n^d = 1$ degradation features, $d = 1, .., 4$, and $n^p = 1$ power features, $p = 1, .., 4$. We consider 4 degradation states for the generators, $s_1^d =$

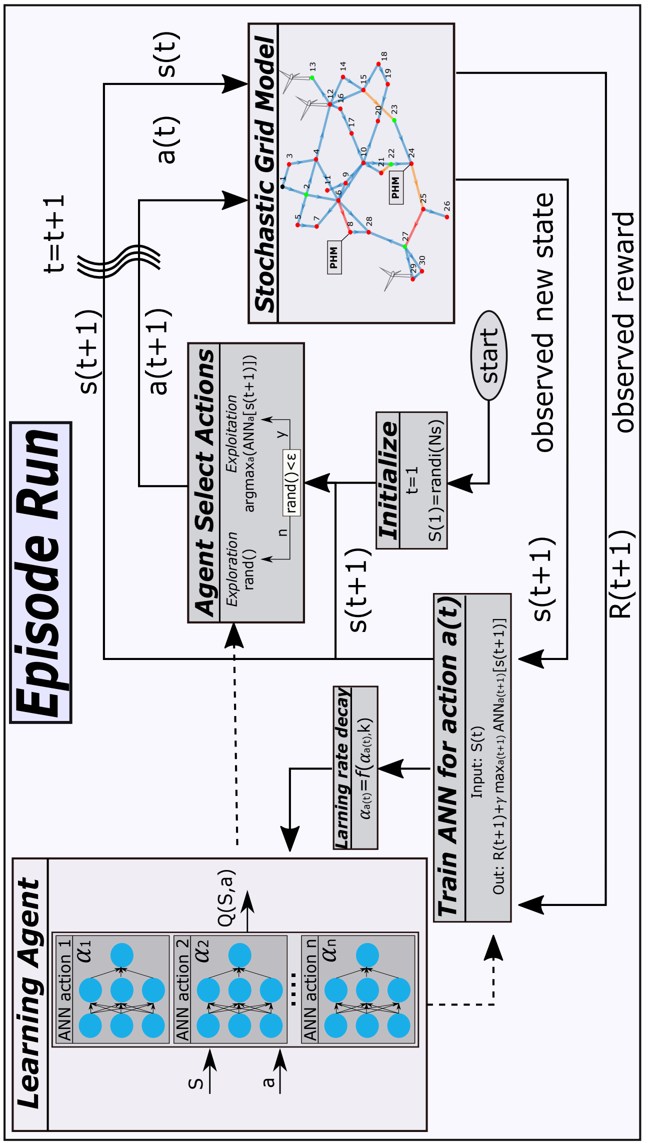

Figure 2: The flow chart displays an episode run and how the learning agent interacts with the environment (i.e. the power grid equipped with PHM devices) in the developed Reinforcement Learning framework; dashed-line arrows indicate when the learning agent takes part in the episode run.

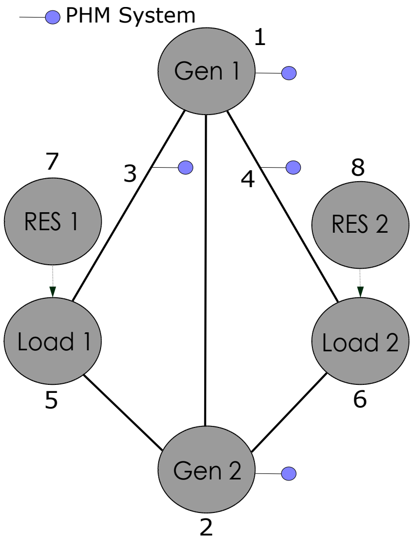

Figure 3: The power grid structure and the position of the 4 PHM-equipped systems, 2 Renewable Energy Sources, 2 loads and 2 controllable generators.

$\{1,..,S_1^d = 4\}$, $d = 1, 2$, whereas the 2 degrading power lines, $d = 3, 4$, have three states: $s_1^d = \{1,..,S_1^d = 3\}$. State 1 refers to the AGAN condition, whereas state S_1^d to the failure state and states $1 < s_1^d < S_1^d$ to degraded states in ascending order. For each load, we consider 3 states of increasing power demand $s_1^p = \{1,..,S_1^p = 3\}$, $p = 5, 6$. Three states of increasing power production are associated to the Renewable Energy Sources, $s_1^p = \{1,..,S_1^p = 3\}, p = 7, 8$. Then, the state vector at time t reads:

$$\mathbf{S}(t) = \left[s_1^1, s_1^2, s_1^3, s_1^4, s_1^5, s_1^6, s_1^7, s_1^8\right]$$

Space $\mathcal{S}$ is made up of 11664 points.

The agent can operate both generators to maximize the system revenues by minimizing the unbalance between demand and production, while preserving the structural and functional integrity of the system. Then, $g \in G = \{1, 2\}$ and $\varrho = 1, ..., |G| = 2$. Being in this case subscript $\varrho = g$, it can be omitted.

Notice that other actions can be performed by other agents on other components (e.g. transmission lines). These are assumed not under the agent control, and, thus, are included in the environment. Then, the action vector reads $\mathbf{a} = [a_1, a_2]$, whereas $A_g = \{1, .., 5\}$, $g \in \{1, 2\}$, and $|\mathcal{A}| = 25$. This gives rise to a 291600 state-action pairs. For each generator, the first 3 (operational) actions concern the power output, which can be set to one out of the three allowed levels. The last 2 actions are preventive and corrective maintenance actions, respectively. CM is mandatory for failed generators.

Highly degraded generators (i.e. $S_g^d = 3,\ d = 1, 2$) can be operated at the lower power output levels, only ($a_g = 1$ action).

Tables 1-3 display, respectively, the costs for each action and the corresponding power output of the generators, the line electric parameters and the relation between states s_1^p and the power variable settings.

3.2. Probabilistic model

We assume that the two loads have identical transition probability matrices and also the degradation of the transmission cables and generators are described by the same Markov process. Thus, for ease of notation, the components subscripts have been dropped.

Each action $\mathbf{a} \in \mathcal{A}$ is associated to a specific transition probability matrix $\mathcal{P}_g^{\mathbf{a}}$, describing the evolution of the generator health state conditioned by its operative state or maintenance action.

The transition matrices for the considered features are reported in Appendix 6. Notice that the probabilities associated to operational actions, namely $a_g = 1, 2, 3$, affect differently the degradation of the component. Moreover, for those actions, the bottom row corresponding to the failed state has only zero entries, indicating that operational actions cannot be taken on failed generators, as only CM is allowed.

3.3. Reward model

The reward is made up of four different contributions: (1) the revenue from selling electric power, (2) the cost of producing electric power by traditional

Table 1: The power output of the 2 generators in [MW] associated to the 5 available actions and action costs in monetary unit [m.u.].

Action:	1	2	3	4	5
$P_{g=1}$ [MW]	40	50	100	0	0
$P_{g=2}$ [MW]	50	60	120	0	0
$C_{a,g}$ [m.u.]	0	0	0	10	500

Table 2: The transmission lines ampacity and reactance proprieties.

From	To	Ampacity [A]	X [Ω]
Gen 1	Load 1	125	0.0845
Gen 1	Load 2	135	0.0719
Gen 1	Gen 2	135	0.0507
Load 1	Gen 2	115	0.2260
Load 2	Gen 2	115	0.2260

Table 3: The physical values of the power settings in [MW] associated to each state S_1^p of component $p \in P$.

	State index s_1^p	1	2	3
$p = 5$	Demanded [MW]	60	100	140
$p = 6$	Demanded [MW]	20	50	110
$p = 7$	Produced [MW]	0	20	30
$p = 8$	Produced [MW]	0	20	60

generators, (3) the cost associated to the performed actions and (4) the cost of not serving energy to the customers. Mathematically, the reward reads:

$$R(\mathbf{S}_t \ , \ \mathbf{a}_t \ , \ \mathbf{S}_{t+1}) =$$

$$= \sum_{p=5}^{6} \Bigg(L_p(t) - \frac{ENS_p(t)}{\Delta_t}) \Bigg) \cdot C_{el} - \sum_{g=1}^{2} P_g \cdot C_g - \sum_{g=1}^{2} C_{a,g} - \sum_{p=5}^{6} ENS_p(t) \cdot C_{ENS}$$

where L_p is the power demanded by element p, C_{el} is the price paid by the loads for buying a unit of electric power, P_g is the power produced by the generators, C_g is the cost of producing the unit of power, $C_{a,g}$ is the cost of action a_g on generator g, $\Delta_t = 1h$ is the time difference between the present and the

next system state and ENS_p is the energy not supplied to load p; this is a function of the grid state $\mathbf{S}$, grid electrical proprieties and availability $\mathcal{M}$, i.e. $ENS(t) = \mathcal{G}(\mathbf{S}, \mathcal{M})$ where $\mathcal{G}$ defines the constrained DC power flow solver ([35], see Figure 2). C_{ENS} is the cost of the energy not supplied.
Costs C_{ENS}, C_g and C_{el} are set to 5, 4 and 0.145 monetary unit (m.u.) per-unit of energy or power, respectively. These values are for illustration, only.

4. Results and discussions

The developed algorithm (pseudo - code 1 in Appendix) provides a non-tabular solution to the stochastic control problem, which is compared to the reference Bellman's optimality (pseudo-code 2 in Appendix). The algorithm runs for $N_{ei} = 1e4$ episodes with truncation window $T = 20$, initial learning rate $\alpha_0 = 0.02$, initial expiration rate $\epsilon_0 = 0.9$ and decay coefficients $K_\alpha = 1e{-}2$. The learning agent is composed of 25 fully-connected ANNs having architectures defined by $\mathbf{N}_{layers} = [8, 10, 5, 1]$, that is: 1 input layer with 8 neurons, 1 output layer with 1 neuron and 2 hidden layers with 10 and 5 neurons, respectively. The results of the analysis are summarized in the top panel in Figure 4, where the curves provide a compact visualization of the distribution of $Q_{\pi^*}(\mathbf{S}, \mathbf{a})$ over the states, for the available 25 combinations of actions. For comparison, the reference optimal action-value function is displayed in the bottom panel. The results of the two algorithms are in good agreement, although minor inevitable approximation errors can be observed for some of the state-action pairs. Three clusters can be identified: on the far left, we find the set of states for which CM on both generators is performed; being CM a costly action, this leads to a negative expectation of the discounted reward. The second cluster (*C 2*) corresponds to the 8 possible combinations of one CM and any other action on the operating generator. The final cluster (*C 1*) of 16 combinations of actions includes only PM and operational actions. If corrective maintenance is not performed, higher rewards are expected.

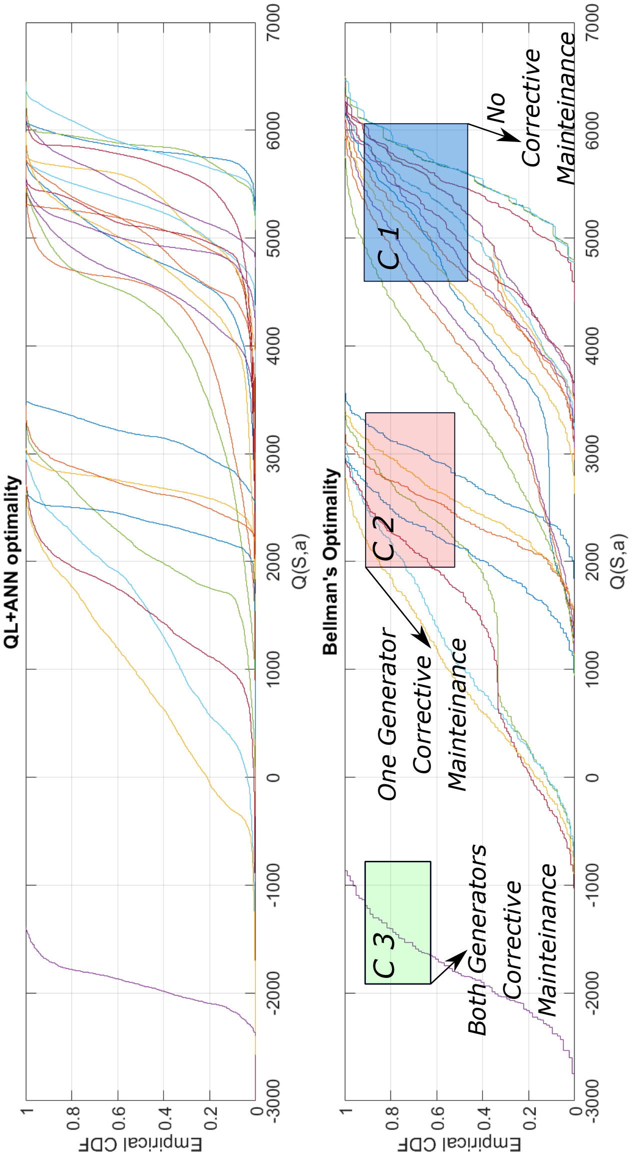

Figure 4: The $Q(s, a)$ values displayed using ECDFs and the 3 clusters. Comparison between the reference Bellman's solution (bottom plot) and the QL+ANN solution (top plot).

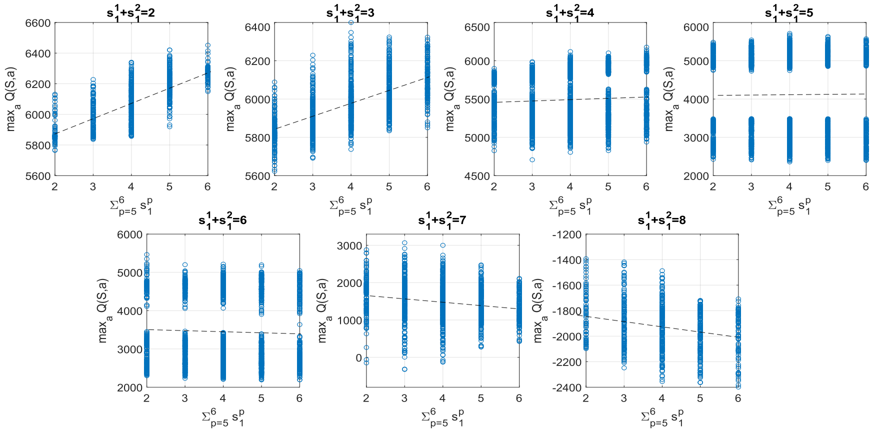

Figure 5: The maximum expected reward, $\hat{q}_a(\mathbf{S}|\boldsymbol{\mu}_a)$, for increasing total load and different degrading condition of the generators.

In Figure 5, each sub-plot shows the maximum expected discounted reward given by the policy found by Algorithm 1, conditional to a specific degradation state of the generators and for increasing electric load demands. It can be noticed that when the generators are both healthy or slightly degraded (i.e. $\sum_{d=1}^{2} s_1^d = 2, 3, 4$), an increment in the overall power demand entails an increment in the expected reward, due to the larger revenues from selling more electric energy to the customers (dashed lines display the average trends). On the other hand, if the generators are highly degraded or failed (i.e. $\sum_{d=1}^{2} s_1^d = 7, 8$), an increment in the load demand leads to a drop in the expected revenue. This is due to the increasing risk of load curtailments and associated cost (i.e. cost of energy not supplied), and to the impacting PM and CM actions costs. Similar results can be obtained solving the Bellman's optimality (e.g. see [36]).

To compare the Q values obtained from Algorithm 1 to the Bellman's reference, a convergence plot for 3 states is provided in Figure 6. Every state is representative of one of the 3 clusters *C 1*, *C 2* and *C 3* (see Figure 4): $\mathbf{S}_1 = [1, 1, 1, 1, 1, 1, 1, 1]$ has both generators in the AGAN state, $\mathbf{S}_2 = [4, 1, 1, 1, 1, 1, 1, 1]$ has one generator out of service while $\mathbf{S}_3 = [4, 4, 3, 3, 3, 3, 3, 3]$ has both generators failed. Figure 6 also reports the corresponding reference Bellman's solutions (dashed lines): their closeness indicates that the Reinforcement Learning algorithm converges to the true optimal policy.

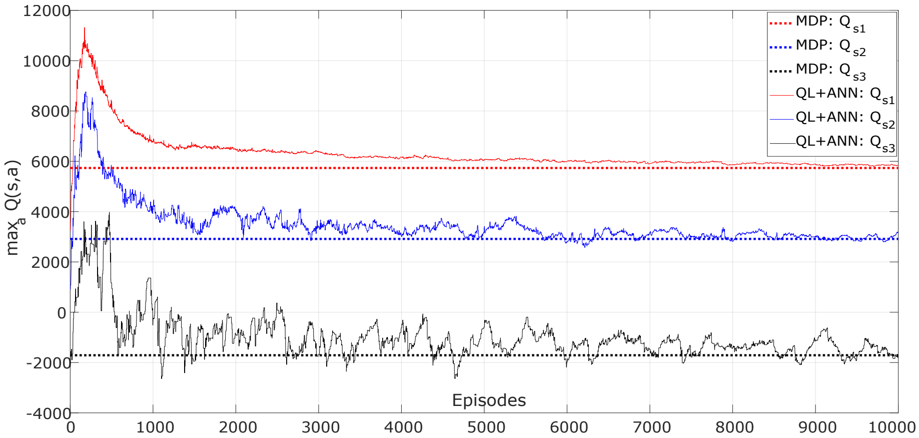

Figure 6: The convergence of the $\max_{a\in\{1,\ldots,|\mathcal{A}|\}} \hat{q}_a(\mathbf{S}|\boldsymbol{\mu}_a)$ for 3 representative system states (i.e. generators fully-operative, partially failed/degraded and fully-failed).

4.1. Policies comparison

Table 4 compares the results obtained from the developed Reinforcement Learning algorithm with the Bellman's optimality and two synthetic policies. The first suboptimal policy is named Q_{rnd}, in which actions are randomly selected. This comparison is used as reference worst case, as it is the policy that a non-expert decision maker would implement on the Power Grid. The second synthetic policy, named Q_{exp}, is based on experience: the agent tries to keep the balance between loads and production by optimally setting the power output of the generators. However, he/she will never take PM actions. This reference policy is that which an agent not informed on the health state of the components would apply on the Power Grid elements.

Table 4 shows on the first row the Normalized Root Mean Squared Error (NRMSE, i.e., the error averaged over all the state action pairs and normalized over the min-max range of the Bellman's Q) between the considered policies and the Bellman's reference Q.

In the next rows, Table 4 shows the averaged non-discounted return $\overline{R}(t) = \frac{\sum_{t=1}^{T} R(t)}{T}$, independent from the initial state of the system, its standard deviation $\sigma[R(t)]$, the average value of the energy not supplied, $\overline{ENS}$, and the corresponding standard deviation $\sigma[ENS]$. These values have been obtained by Monte Carlo, simulating the system operated according to the four considered policies.

We can see that the Reinforcement Learning policy yields negative values of the average energy-not-supplied (about -45.2 MW), which are smaller than those of the reference Bellman's policy solution method (-6.7 MW). This indicates that both Bellman's and Reinforcement Learning policies yield an overproduction of electric power. However, the reward of the Bellman's solution is larger, due to the closer balance between load and demand and, thus, lower costs associated to the overproduction.

Concerning the expert-based policy Q_{exp}, it behaves quite well in term of average ENS, with results comparable to the Bellman's optimality. On the other hand, the resulting Q and $\overline{R}(t)$ are both smaller than those of the Bellman's policy and the Reinforcement Learning policy. This is due to the increased occurrence frequency of CM actions and associated costs. The random policy produces sensibly worsen the results of both ENS and rewards.

To further explain these results, we can look at Table 5. For the four considered policies, the panels report the frequency of selection of the 5 actions available for the generators, conditional to their degradation state: the Bellman's policy in the top panel, left-hand side, the Reinforcement Learning policy in the top panel, right-hand side, the suboptimal random policy in the bottom panel, left-hand side, and the expert-based policy in the bottom panel, right-hand side. In each panel, the first 4 rows refer to the possible degradation states of Gen 1, whilst the last 4 rows show the results for Gen 2.

With respect to the Bellman solution it can be observed that when Gen 1 is nearly failed ($s_1^1 = 3$), it undergoes PM for the vast majority of the scenarios

(80.9 % of the states). Conversely, when Gen 2 is nearly failed ($s_1^2 = 3$), the optimal policy is more inclined to keep it operating (54.3 % of the scenarios) rather than perform a PM (45.7 %). This means that in the states for which $s_1^2 = 3$, the agent is ready to: (1) take the risk of facing failure and (2) have the generator forcefully operated at the minimum power regime. This difference in the operation of the two generators can be explained by considering the specific topology of the system, the inherent asymmetry in the load, renewable and controllable generators capacity, and the PHM devices which are not uniformly allocated on the grid.

In terms of action preferences, the Reinforcement Learning policy presents some similarities and differences when compared to the Bellman ones. In particular, given a nearly failed state for Gen 1, this is more likely to undergo PM (20.4 % of the times) if compared to Gen 2 (only 14.1 %). This is in line with the results of the Bellman's policy. However, a main difference can be pointed out: following the Reinforcement Learning policy, PM actions are taken less frequently, with a tendency to keep operating the generators. This is reflected in the rewards, which are slightly smaller. Nonetheless, the Reinforcement Learning policy tends to optimality and greatly outperforms the random policy, as expected, and also presents an improvement with respect to the expert-based solution to the optimization problem. This gives evidence of the benefit of PHM on the Power Grid.

As expected, the action selection frequencies of the randomized policy do not depend on the states of the generators and PM are not selected in the expert-based policy, as required when it has been artificially generated.

One main drawback of the developed algorithm is that it is computationally quite intensive (approximately 14 hours of calculations on a standard machine, last row in Table 4). This is due to the many ANNs trainings, which have to be repeated for each reward observation. However, its strength lies in its applicability to high dimensional problems and with continuous states. Furthermore, its effectiveness has been demonstrated by showing that the derived optimal policy greatly outperformed an alternative non-optimal strategy, with expected rewards comparable to the true optimality. Further work will be dedicated to reducing the computational time needed for the analysis, possibly introducing time-saving training algorithms and efficient regression tools.

Table 4: Comparison between the policy derived from the QL+ANN Algorithm 1, a synthetic non-optimal random policy, an expert-based policy and the reference Bellman's optimality

Policy π	Bellman's	QL+ANN	Q_{rnd}	Q_{exp}
NRMSE	0	0.083	0.35	0.11
$\overline{R}(t)$	532.9	439.1	260.3	405.2
$\sigma[R(t)]$	347.5	409.3	461.6	412.2
$\overline{ENS}$	-6.71	-45.22	15.16	-8.1
σ[ENS]	71.2	75.8	80.9	66.2
Comp. time [s]	17.3e4	5e4	-	-

5. Discussion, limitations and future directions

The proposed framework has been tested on a scaled-down power grid case study with discrete states and relatively small number of actions. This was a first necessary step to prove the effectiveness of the method by comparison with a true optimal solution (i.e., the Bellman's optimal solution). It is worth remarking that RL cannot learn from direct interaction with the environment, as this would require unprofitably operating a large number of systems. Then, a realistic simulator of the state evolution depending on the actions taken is required. This seems not a limiting point in the Industry 4.0 era, when digital twins are more and more common and refined. Future research efforts will be devoted to test the proposed framework on numerical models of complex systems (for which reference Bellman's solution is not obtainable) and on empirical data, collected from real world systems, is also expected.

Table 5: Decision-maker actions preferences. Percentage of actions taken on the generators conditional to their degradation state (following the Bellman's policy, the Reinforcement Learning policy, the sub-optimal policy and the expert-based policy).

	Bellman's policy					Reinforcement Learning policy				
$a_1 =$	1	2	3	4	5	1	2	3	4	5
$s_1^1 = 1$	24.3	7.4	58	10.2	0	7.5	20.5	71.5	0.65	0
$s_1^1 = 2$	28.2	6.4	65.4	0	0	0.6	29.4	69.4	0.6	0
$s_1^1 = 3$	19.1	0	0	80.9	0	79.6	0	0	20.4	0
$s_1^1 = 4$	0	0	0	0	100	0	0	0	0	100
$a_2 =$	1	2	3	4	5	1	2	3	4	5
$s_1^2 = 1$	38.9	8.6	45	7.4	0	2.7	27.6	69.6	0	0
$s_1^2 = 2$	36.1	11.4	52.5	0	0	2.4	24.3	72.9	0.3	0
$s_1^2 = 3$	54.3	0	0	45.7	0	85.9	0	0	14.1	0
$s_1^2 = 4$	0	0	0	0	100	0	0	0	0	100

	Randomized Policy					Expert-based policy				
$a_1 =$	1	2	3	4	5	1	2	3	4	5
$s_1^1 = 1$	25.6	25.2	24.6	24.3	0	0	37	63	0	0
$s_1^1 = 2$	23.8	25.3	25	25.9	0	0	37	63	0	0
$s_1^1 = 3$	52.1	0	0	47.9	0	100	0	0	0	0
$s_1^1 = 4$	0	0	0	0	100	0	0	0	0	100
$a_2 =$	1	2	3	4	5	1	2	3	4	5
$s_1^2 = 1$	24.6	24.9	25.6	24.7	0	76	2.4	21.6	0	0
$s_1^2 = 2$	24.5	25.1	24.9	25.4	0	76.6	1.8	21.6	0	0
$s_1^2 = 3$	50.4	0	0	49.6	0	100	0	0	0	0
$s_1^2 = 4$	0	0	0	0	100	0	0	0	0	100

6. Conclusion

A Reinforcement Learning framework for optimal O&M of power grid system under uncertainty is proposed. A method which combines Q-learning algorithm and an ensemble of Artificial Neural Networks is developed, which is applicable to large systems with high dimensional state-action spaces. An analytical (Bellman's) solution is provided for scaled-down power grid, which includes Prognostic Health Management devices, renewable generators and degrading components, giving evidence that Reinforcement Learning can really exploit the information gathered from Prognostic Health Management devices, which helps to select optimal O&M actions on the system components. The proposed strategy provides accurate solutions comparable to the true optimal. Although inevitable approximation errors have been observed and computational time is an open issue, it provides useful direction for the system operator. In fact, he/she can now discern whether a costly repairing action is likely to lead to a long-term economic gain or is more convenient to delay the maintenance.

References

[1] J. Dai, Z. D. Wang, P. Jarman, Creepage discharge on insulation barriers in aged power transformers, IEEE Transactions on Dielectrics and Electrical Insulation 17 (4) (2010) 1327–1335. doi:10.1109/TDEI.2010.5539705.

[2] R. Goyal, B. K. Gandhi, Review of hydrodynamics instabilities in francis turbine during off-design and transient operations, Renewable Energy 116 (2018) 697 – 709. doi:https://doi.org/10.1016/j.renene.2017.10.012.
URL http://www.sciencedirect.com/science/article/pii/S0960148117309734

[3] H. Aboshosha, A. Elawady, A. E. Ansary, A. E. Damatty, Review on dynamic and quasi-static buffeting response of transmission lines under synoptic and non-synoptic winds, Engineering Structures 112 (2016) 23 –

46. doi:https://doi.org/10.1016/j.engstruct.2016.01.003.
URL http://www.sciencedirect.com/science/article/pii/S0141029616000055

[4] M. Compare, L. Bellani, E. Zio, Optimal allocation of prognostics and health management capabilities to improve the reliability of a power transmission network, Reliability Engineering & System Safetydoi:https://doi.org/10.1016/j.ress.2018.04.025.
URL http://www.sciencedirect.com/science/article/pii/S0951832017306816

[5] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, Y. Wang, Review of road traffic control strategies, Proceedings of the IEEE 91 (12) (2003) 2043–2067. doi:10.1109/JPROC.2003.819610.

[6] J. Jin, X. Ma, Hierarchical multi-agent control of traffic lights based on collective learning, Engineering Applications of Artificial Intelligence 68 (2018) 236 – 248. doi:https://doi.org/10.1016/j.engappai.2017.10.013.
URL http://www.sciencedirect.com/science/article/pii/S0952197617302658

[7] R. Yousefian, R. Bhattarai, S. Kamalasadan, Transient stability enhancement of power grid with integrated wide area control of wind farms and synchronous generators, IEEE Transactions on Power Systems 32 (6) (2017) 4818–4831. doi:10.1109/TPWRS.2017.2676138.

[8] M. S. Mahmoud, N. M. Alyazidi, M. I. Abouheaf, Adaptive intelligent techniques for microgrid control systems: A survey, International Journal of Electrical Power & Energy Systems 90 (2017) 292 – 305. doi:https://doi.org/10.1016/j.ijepes.2017.02.008.
URL http://www.sciencedirect.com/science/article/pii/S0142061516325042

[9] E. Kuznetsova, Y.-F. Li, C. Ruiz, E. Zio, G. Ault, K. Bell, Reinforcement learning for microgrid energy management, Energy 59 (2013) 133 – 146.

doi:https://doi.org/10.1016/j.energy.2013.05.060.
URL http://www.sciencedirect.com/science/article/pii/S0360544213004817

[10] M. Compare, P. Marelli, P. Baraldi, E. Zio, A markov decision process framework for optimal operation of monitored multi-state systems, Proceedings of the Institution of Mechanical Engineers Part O Journal of Risk and Reliability.

[11] R. S. Sutton, D. Precup, S. Singh, Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning, Artificial Intelligence 112 (1) (1999) 181 – 211. doi:https://doi.org/10.1016/S0004-3702(99)00052-1.
URL http://www.sciencedirect.com/science/article/pii/S0004370299000521

[12] C. Szepesvari, Algorithms for Reinforcement Learning, Morgan and Claypool Publishers, 2010.

[13] T. I. Ahamed, P. N. Rao, P. Sastry, A reinforcement learning approach to automatic generation control, Electric Power Systems Research 63 (1) (2002) 9 – 26. doi:https://doi.org/10.1016/S0378-7796(02)00088-3.
URL http://www.sciencedirect.com/science/article/pii/S0378779602000883

[14] J. .A, I. Ahamed, J. R. V. P., Reinforcement learning solution for unit commitment problem through pursuit method 0 (2009) 324–327.

[15] M. Glavic, D. Ernst, L. Wehenkel, A reinforcement learning based discrete supplementary control for power system transient stability enhancement, in: Engineering Intelligent Systems for Electrical Engineering and Communications, 2005, pp. 1–7.

[16] R. Lu, S. H. Hong, X. Zhang, A dynamic pricing demand response algorithm for smart grid: Reinforcement learning ap-

proach, Applied Energy 220 (2018) 220 – 230. doi:https://doi.org/10.1016/j.apenergy.2018.03.072.
URL http://www.sciencedirect.com/science/article/pii/S0306261918304112

[17] E. Jasmin, T. I. Ahamed, V. J. Raj, Reinforcement learning approaches to economic dispatch problem, International Journal of Electrical Power & Energy Systems 33 (4) (2011) 836 – 845. doi:https://doi.org/10.1016/j.ijepes.2010.12.008.
URL http://www.sciencedirect.com/science/article/pii/S014206151000222X

[18] V. Nanduri, T. K. Das, A reinforcement learning model to assess market power under auction-based energy pricing, IEEE Transactions on Power Systems 22 (1) (2007) 85–95. doi:10.1109/TPWRS.2006.888977.

[19] J. G. Vlachogiannis, N. D. Hatziargyriou, Reinforcement learning for reactive power control, IEEE Transactions on Power Systems 19 (3) (2004) 1317–1325. doi:10.1109/TPWRS.2004.831259.

[20] D. Ernst, M. Glavic, F. Capitanescu, L. Wehenkel, Reinforcement learning versus model predictive control: A comparison on a power system problem, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39 (2) (2009) 517–529. doi:10.1109/TSMCB.2008.2007630.

[21] J. R. Vzquez-Canteli, Z. Nagy, Reinforcement learning for demand response: A review of algorithms and modeling techniques, Applied Energy 235 (2019) 1072 – 1089. doi:https://doi.org/10.1016/j.apenergy.2018.11.002.
URL http://www.sciencedirect.com/science/article/pii/S0306261918317082

[22] H. Li, T. Wei, A. Ren, Q. Zhu, Y. Wang, Deep reinforcement learning: Framework, applications, and embedded implementations: Invited paper,

in: 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2017, pp. 847–854. doi:10.1109/ICCAD.2017.8203866.

[23] J. Wu, H. He, J. Peng, Y. Li, Z. Li, Continuous reinforcement learning of energy management with deep q network for a power split hybrid electric bus, Applied Energy 222 (2018) 799 – 811. doi:https://doi.org/10.1016/j.apenergy.2018.03.104.
URL http://www.sciencedirect.com/science/article/pii/S0306261918304422

[24] R. Lu, S. H. Hong, Incentive-based demand response for smart grid with reinforcement learning and deep neural network, Applied Energy 236 (2019) 937 – 949. doi:https://doi.org/10.1016/j.apenergy.2018.12.061.
URL http://www.sciencedirect.com/science/article/pii/S0306261918318798

[25] T. Sogabe, D. B. Malla, S. Takayama, S. Shin, K. Sakamoto, K. Yamaguchi, T. P. Singh, M. Sogabe, T. Hirata, Y. Okada, Smart grid optimization by deep reinforcement learning over discrete and continuous action space, in: 2018 IEEE 7th World Conference on Photovoltaic Energy Conversion (WCPEC) (A Joint Conference of 45th IEEE PVSC, 28th PVSEC 34th EU PVSEC), 2018, pp. 3794–3796. doi:10.1109/PVSC.2018.8547862.

[26] J. Zhang, C. Lu, J. Si, J. Song, Y. Su, Deep reinforcement leaming for short-term voltage control by dynamic load shedding in china southem power grid, in: 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1–8. doi:10.1109/IJCNN.2018.8489041.

[27] D. O'Neill, M. Levorato, A. Goldsmith, U. Mitra, Residential demand response using reinforcement learning, in: 2010 First IEEE International Conference on Smart Grid Communications, 2010, pp. 409–414. doi:10.1109/SMARTGRID.2010.5622078.

[28] M. Glavic, R. Fonteneau, D. Ernst, Reinforcement learning for electric power system decision and control: Past considerations and perspectives,

IFAC-PapersOnLine 50 (1) (2017) 6918 – 6927, 20th IFAC World Congress. doi:https://doi.org/10.1016/j.ifacol.2017.08.1217.
URL http://www.sciencedirect.com/science/article/pii/S2405896317317238

[29] F. Cannarile, M. Compare, P. Baraldi, F. Di Maio, E. Zio, Homogeneous continuous-time, finite-state hidden semi-markov modeling for enhancing empirical classification system diagnostics of industrial components, Machines 6. doi:10.3390/machines6030034.

[30] R. S. Sutton, A. G. Barto, Reinforcement learning i: Introduction (2017).

[31] M. Riedmiller, Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method, in: European Conference on Machine Learning, Springer, 2005, pp. 317–328.

[32] B. D. Ripley, Pattern recognition and neural networks, Cambridge university press, 2007.

[33] S. S. Haykin, S. S. Haykin, S. S. Haykin, S. S. Haykin, Neural networks and learning machines, Vol. 3, Pearson Upper Saddle River, NJ, USA:, 2009.

[34] J. Frank, S. Mannor, D. Precup, Reinforcement learning in the presence of rare events, in: Proceedings of the 25th international conference on Machine learning, ACM, 2008, pp. 336–343.

[35] R. Rocchetta, E. Patelli, Assessment of power grid vulnerabilities accounting for stochastic loads and model imprecision, International Journal of Electrical Power & Energy Systems 98 (2018) 219 – 232. doi:https://doi.org/10.1016/j.ijepes.2017.11.047.
URL http://www.sciencedirect.com/science/article/pii/S0142061517313571

[36] R. Rocchetta, M. Compare, E. Patelli, E. Zio, A reinforcement learning framework for optimisation of power grid operations and maintenance, in: Reliable engineering computing, REC 2018, 2018.

[37] E. Gross, On the bellmans principle of optimality, Physica A: Statistical Mechanics and its Applications 462 (2016) 217 – 221. doi:https://doi.org/10.1016/j.physa.2016.06.083.
URL http://www.sciencedirect.com/science/article/pii/S037843711630351X

Appendix 1

Formally, a MDP is a tuple $\langle S$,A,R,$\mathcal{P}\rangle$, where S is a finite state set, $A(s)$ is a finite action set with $s \in S$, R is a reward function such that $R(s,a) \in \mathbb{R}, \forall s \in S, a \in \mathcal{A}$ and P is a probability function mapping the state action space:

$$\mathcal{P}_{s,a,s'} : S \times A \times S \mapsto [0,1]$$

A specific policy π is defined as a map from the state space to the action space $\pi : S \mapsto A$ with $\pi(s) \in A(s)$ $\forall s \in S$ and it belongs to the set of possible policies Π. The action-value function $Q_\pi(s,a)$ is mathematically defined as [**30**]:

$$Q_\pi(s,a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) | S_0 = s, A_0 = \pi(s_0) \right] \; s \in S$$

where $\gamma \in [0,1]$ is the discount factor and a $\gamma < 1$ is generally employed to avoid divergence of the cumulative rewards as well as to reflect the fact that is some cases earlier rewards are more valuable than future rewards. The Bellman's optimality equation provides an analytical expression for $Q_{\pi^*}(s,a)$, which is the action-value function for optimal policy π^*. The Bellman's optimality is defined by a recursive equation as follows [37]-[30]:

$$Q_{\pi^*}(s_t, a_t) = \sum_{s_{t+1}} \mathcal{P}(s_{t+1}|s_t, a_t) \left[R(s_{t+1}, a_t, s_t) + \underset{a_{t+1}}{max} \; \gamma Q_{\pi^*}(s_{t+1}, a_{t+1}) \right] \quad (7)$$

Equation 7 can be solved by Dynamic Programming such as policy iteration or value iteration [30].

The QL+ANN algorithm 1 consists of two phases: (1) an initialization phase of the ANNs ensemble and (2) the learning phase, where Q-learning algorithm is

Algorithm 1 The QL+ANN Algorithm.

```
Set $ei = 1$, $n_{ei}$ $N_{ei}$, $K_\alpha$, $\epsilon_0$, $\alpha_0$, $\gamma$, $\mathbf{N}_{layers}$;
**Phase 1: Off-Line Training**
Initialize Networks $\mathcal{N}_l$ and $t_l = 1$, $l = 1, ...|\mathcal{A}|$ with architecture $\mathbf{N}_{layers}$;
Sample transitions $\mathbf{S}_t, \mathbf{a}_t \rightarrow \mathbf{S}_{t+1}, \mathbf{a}_{t+1}$ and observe rewards $R_t$, $t = 1, ..., n_{ei}$;
Approximate $Q$ by the MC estimate $Y_t = \sum_{t'=t}^{t+\Phi} \gamma^{t'-t} \cdot R_{t'}$
Train each $\mathcal{N}_l$ using $\{\mathbf{S}_t | t = 1, ..., n_{ei}, \mathbf{a}_t = l\}$ and the estimated $Y_t$ (output);

**Phase 2: Learning**
**while** $ei < N_{ei}$ (Episodic Loop) **do**
    Set $t = 1$ initialize state $\mathbf{S}_t$ randomly
    $\epsilon = \epsilon_0 \cdot \tau_\epsilon^{ei}$
    **while** $t < T$ (episode run) **do**
          **if** $rand() < 1 - \epsilon$ (exploit)
             $\mathbf{a}_t = arg \max_{l \in \{1,...,|\hat{A}_{g_\varrho}|\}} \hat{q}_l(\mathbf{S}_t | \boldsymbol{\mu}_l)$
          **else** (explore)
             Select $\mathbf{a}_t$ randomly s.t. $\mathbf{a}_t \in \hat{A}_{g_\varrho}$
          **end**
        Take action $\mathbf{a}_t$, observe $\mathbf{S}_{t+1}$ and reward $R_t$
        **Update** network $\mathcal{N}_{\mathbf{a}_t}$ weights, $\epsilon$ and $\alpha$
        $\boldsymbol{\mu}_{\mathbf{a}_t} \leftarrow \boldsymbol{\mu}_{\mathbf{a}_t} + \alpha_{\mathbf{a}_t} \cdot [R_t + \gamma \cdot \max_{l \in \{1,...,|\mathcal{A}|\}} \hat{q}_l(\mathbf{S}_{t+1} | \boldsymbol{\mu}_l) - \hat{q}_{\mathbf{a}_t}(\mathbf{S}_t | \boldsymbol{\mu}_{\mathbf{a}_t})] \cdot \nabla \hat{q}_{\mathbf{a}_t}(\mathbf{S}_t | \boldsymbol{\mu}_{\mathbf{a}_t})$
        $\alpha_{\mathbf{a}_t} = \alpha_0 \cdot (\frac{1}{1 + K_\alpha \cdot t_{\mathbf{a}_t}})$
        Set $t = t + 1$ and $t_{\mathbf{a}_t} = t_{\mathbf{a}_t} + 1$
    **end while**
    go to next episode $ei = ei + 1$
**end while**
```

used in combination to the ANNs to learn an optimal decision-making policy.In phase (1) an ANN is associated with each action vector **a** and its architecture, i.e. number of layers and nodes per layer, is defined by the $\mathbf{N}_{layers}$ vector. Each network is first trained using the Levenberg-Marquardt algorithm, providing as input the state vectors and as output the estimator of Q obtained from the future rewards. In phase (2) the Reinforcement Learning algorithm run, Artificial Neural Networks select the actions and the ensemble is incrementally trained to improve its predictive performance. Notice that, whilst tabular Reinforcement Learning methods are guaranteed to convergence to an optimal action-value function for a Robbins-Monro sequence of step-sizes α_t, a generalized convergence guarantee for non-tabular methods has not been provided yet and an inadequate setup can lead to suboptimal, oscillating or even diverging solutions.Thus, an empirical convergence test has been designed to assess the the reliability of the results. For further details, please refer to [30].

Appendix 2

$$\mathcal{P}_d^{a_d=1} = \begin{bmatrix} 0.98 & 0.02 & 0 & 0 \\ 0 & 0.95 & 0.05 & 0 \\ 0 & 0 & 0.9 & 0.1 \\ - & - & - & - \end{bmatrix} \; d = 1,2 \quad \mathcal{P}_d^{a_d=2} = \begin{bmatrix} 0.97 & 0.03 & 0 & 0 \\ 0 & 0.95 & 0.05 & 0 \\ - & - & - & - \\ - & - & - & - \end{bmatrix} \; d = 1,2$$

$$\mathcal{P}_d^{a_d=3} = \begin{bmatrix} 0.95 & 0.04 & 0.01 & 0 \\ 0 & 0.95 & 0.04 & 0.01 \\ - & - & - & - \\ - & - & - & - \end{bmatrix} \; d = 1,2$$

$$\mathcal{P}_d^{a_d=4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0.5 \\ - & - & - & - \end{bmatrix} d = 1,2 \qquad \mathcal{P}_d^{a_d=5} = \begin{bmatrix} - & - & - & - \\ - & - & - & - \\ - & - & - & - \\ 0.15 & 0 & 0 & 0.85 \end{bmatrix} d = 1,2$$

$$\mathcal{P}_d^{\mathbf{a}} = \begin{bmatrix} 0.9 & 0.08 & 0.02 \\ 0 & 0.97 & 0.03 \\ 0.1 & 0 & 0.9 \end{bmatrix} \forall\ \mathbf{a}, d = 3,4 \qquad \mathcal{P}_p^{\mathbf{a}} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.4 \\ 0.2 & 0.4 & 0.4 \end{bmatrix} \forall\ \mathbf{a}, p = 5,6$$

$$\mathcal{P}_7^{\mathbf{a}} = \begin{bmatrix} 0.5 & 0.1 & 0.4 \\ 0.3 & 0.3 & 0.4 \\ 0.1 & 0.4 & 0.5 \end{bmatrix} \forall\ \mathbf{a} \qquad \mathcal{P}_8^{\mathbf{a}} = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.4 & 0.4 & 0.2 \\ 0 & 0.5 & 0.5 \end{bmatrix} \forall\ \mathbf{a}$$

Algorithm 2 The value iteration algorithm (Bellman's optimality)

```
Initialize $Q$ arbitrarily (e.g. $Q(s,a) = 0\ \forall s \in \mathcal{S}, a \in \mathcal{A}$)
Define tolerance error $\theta \in \mathbb{R}^+$ and $\Delta = 0$
while $\Delta \geq \theta$ do
    for each $s \in \mathcal{S}$ do
        get constrained action set $\mathcal{A}_s$ in $s$
        for each $a \in \mathcal{A}_s$ do
            $q = Q(s,a)$
            $Q(s,a) = \sum_{s'} \mathcal{P}(s'|s,a) \left[ R(s',a,s) + \underset{a'}{max}\ \gamma Q(s',a') \right]$
            $\Delta = \max(\Delta, |q - Q(s,a)|)$
        end for
    end for
end while
Output a deterministic policy $\pi \approx \pi^*$
$\pi(s) = arg \max_{a \in \mathcal{A}_s} Q(s,a)\ \forall s \in \mathcal{S}$
```