





Article

Optimising a Microgrid System by Deep Reinforcement Learning Techniques

David Domínguez-Barbero ^{*}, Javier García-González , Miguel A. Sanz-Bobi  and Eugenio F. Sánchez-Úbeda 

Institute for Research in Technology (IIT), ICAI School of Engineering, Comillas Pontifical University, 28015 Madrid, Spain; javiergg@comillas.edu (J.G.-G.); masanz@comillas.edu (M.A.S.-B.); eugenio.sanchez@comillas.edu (E.F.S.-Ú.)

^{*} Correspondence: david.dominguez@iit.comillas.edu

Received: 16 April 2020; Accepted: 28 May 2020; Published: 2 June 2020



Abstract: The deployment of microgrids could be fostered by control systems that do not require very complex modelling, calibration, prediction and/or optimisation processes. This paper explores the application of Reinforcement Learning (RL) techniques for the operation of a microgrid. The implemented Deep Q-Network (DQN) can learn an optimal policy for the operation of the elements of an isolated microgrid, based on the interaction agent-environment when particular operation actions are taken in the microgrid components. In order to facilitate the scaling-up of this solution, the algorithm relies exclusively on historical data from past events, and therefore it does not require forecasts of the demand or the renewable generation. The objective is to minimise the cost of operating the microgrid, including the penalty of non-served power. This paper analyses the effect of considering different definitions for the state of the system by expanding the set of variables that define it. The obtained results are very satisfactory as it can be concluded by their comparison with the perfect-information optimal operation computed with a traditional optimisation model, and with a Naive model.

Keywords: machine learning; microgrids; optimisation methods; power systems; reinforcement learning

1. Introduction

The transformation of the electric power industry to reduce carbon emissions is changing the roles of both generators and consumers. On the one hand, the generation system is hosting larger amounts of renewable energy sources (RES), with an increasing share of distributed generation, i.e., small generators placed at the distribution level. On the other hand, end consumers are no longer passive agents of the system, but rather active ones that can modulate their consumption during the day, and reduce their net energy balance by means of more efficient and smarter operation of the loads. In this increasingly decentralised power system, involved agents must make decisions continuously in a limited information environment. This decentralisation can be materialised by the creation of micro-networks (microgrids) that are groupings of distributed generation resources, consumption and storage systems that can be partially controlled and that can work both connected to the conventional network or in an isolated manner. The development of smart grids suggests that despite the existing barriers, the penetration of these microgrids will increase considerably soon [1].

Finding the optimal operation of the different elements of the microgrid is not a simple task, and there are several possible approaches in the literature such as robust optimisation [2], stochastic optimisation [3], Mixed Integer-Linear Programming (MILP) combined with a Particle Swarm Optimisation [4] or Reinforcement Learning (RL) [5]. It is important to highlight that robust or stochastic optimisation models require an explicit representation of the microgrid, a mathematical

formulation that includes the objective function to be optimised and the set of all the constraints that define the feasible domain. In addition, the input parameters required to forecast the random variables of interest, such as loads and RES generation, are also a practical need for those optimisation approaches. Therefore, the development of an optimisation-based controller requires the precise selection and estimation of all the parameters and models used to represent each component. Obtaining such information could be extremely difficult in the case of a massive deployment of microgrids. For instance, in a hypothetical future scenario of the power system hosting millions of microgrids (from residential prosumers to larger microgrids in industrial areas), the diversity of consumption patterns and the intermittent nature of renewable generation might represent a serious barrier, as each specific microgrid may require different parameters and even different models [6] leading to unacceptable implementation costs.

In this context, the main advantage of RL is that it allows making decisions in a limited information framework based on the actions taken in the past, and on the observed effects of such actions by automatically adapting the control strategy over time. Unlike standard model-based control approaches, RL techniques do not require the identification of the system in the form of a model. In this way, RL is able to learn from the experience similarly to human behaviour. For example, in [7], it is shown through numerical examples that the proposed learning algorithms can work effectively despite not having prior information on the dynamics of the system. Another example can be seen in [8] where an RL algorithm is proposed to coordinate the loading and unloading of several storage systems with different characteristics. The authors in [9] present a comprehensive review of the main research works found in the literature where RL is applied to a variety of typical problems of power systems, and it presents an RL framework to operate real systems meanwhile a simulation model is used to learn.

Over the last years, a new version of RL methods has emerged. It is called Deep Reinforcement Learning (DRL) because it combines RL with Deep Learning (DL). These methods have an important impact on the academic field, but the industrial applications are currently at the early stages of development. As this is a very active research field, there is an increasing trend of papers that study the application of DRL to power systems. For instance, in [10] several research works that apply RL to power systems are discussed, and some of them are focused on DRL methods. Among them, Deep Q-Network is the most popular technique given its simplicity in contrast to other more complex DRL methods. It should be noted that despite being a recent publication, none of the papers reviewed in [10] address the problem of finding the optimal steady state operation of a microgrid by using DRL.

A more updated and very recent survey paper is [11], where the authors present a broad review of RL applications to power systems, although the categorisation differs from [10]. In particular, one of the categories is the energy supply side, which is very related to the topic studied in this paper, i.e., the microgrid optimal scheduling problem. Under this research area, reference [12] should be highlighted as it addresses the optimisation of the operation of the microgrid using a Deep Q-Network (DQN). The Artificial Neural Network (ANN) used in [12] to implement the DQN is a Multi-Layer Perceptron (MLP). Nevertheless, there are more advanced ANN architectures such as Convolutional Neural Networks (CNN), with one-dimension convolution layers, and Recurrent Neural Networks (RNN), with memory cell layers. These architectures are very suitable for improving the train phase when the input data has a time series structure. For instance, the authors in [5] developed a CNN to find the optimal operation of a microgrid that could be operated both isolated or connected to grid.

The microgrid can be modelled as a Markov Decision Process (MDP). Therefore, beyond the selection of the ANN architecture, the application of DRL (and also RL) requires a definition of a state space of the environment so that the agent can select the optimal actions based on which state the system is in. However, deciding what defines the state of the system may not be trivial. For example, in chess, the position of all the pieces on the chessboard at any given time defines the state of the system, and it is perfectly observable by the player. However, in a microgrid, the decision of what defines the state of the system is not so direct, since the number of variables to be measured can be as

large as desired and the evolution over time is subject to uncertainty. The work in [13] presents the theoretical consequences of varying the configuration of the state and discusses the bias-overfitting tradeoff of DRL. This paper delves also on the effect of the state configuration, and following the suggested research guidelines of [5], the impact of different state definitions of the microgrid are analysed. Therefore this paper takes [5] as a starting point using a DQN based on CNN. In addition, as the time window used to define the state can have different lengths, the main contribution of this paper is the comparative analysis of the impact of the duration of such time window on the performance of the model. In addition, the considered microgrid is more complex than the one studied in [5] as it includes a diesel generator that can act as backup power. Finally, another contribution of this paper is that the economic assessment of the operation of the batteries does not require fixing any price, as the opportunity cost is properly modelled in the sense that the charge/discharge decisions have a direct impact on the usage of the diesel generator. Therefore, this approach is more realistic and can make its application easier to any particular system.

The paper is organised as follows. Section 2 introduces the microgrid elements, and also the modelling fundamentals. The DRL algorithm is described in Section 3, and Section 4 presents its application to a study case where the microgrid structure is described in detail. Section 5 analyses the RL behaviour, comparing the results among the different state configurations. Finally, the main conclusions are summarised in Section 6.

2. The Microgrid Framework

2.1. General Overview of a Microgrid

Figure 1 shows a schematic representation of a microgrid that can be connected to the network through the Point of Common Coupling (PCC). The arrows indicate the possible directions of power flows. The control system is in charge of determining the operation of all the manageable elements of the microgrid. The distributed generation can be fossil fuel generators (for example diesel generators), or renewable sources such as wind or solar generation. Regarding the loads, the microgrid includes elements that consume electricity, such as lighting devices, heating and cooling systems of buildings or charging of electric vehicles. Electricity consumption can change over time and, if the loads are controllable, be susceptible to demand management programs. Finally, the storage systems such as electric batteries, flywheels, heat storage and more, can allow the microgrid to perform multiple functions. Depending on the technology used, the storage capacity can vary and can have a fast or slow dynamic response. In addition, thanks to the use of power electronics and control devices, the storage systems can provide ancillary services such as frequency regulation, and voltage control, although this is out of the scope of this paper. Under a steady-state perspective where the objective is to find the hourly scheduling of all the elements, the primary function of the storage systems is to absorb the inherent variability of electricity consumption and RES generation and to provide backup generation when needed. Finally, in this paper, it is assumed that the microgrid is not connected to the network, and the elements with the dashed line in Figure 1 are not included in the studied configuration.

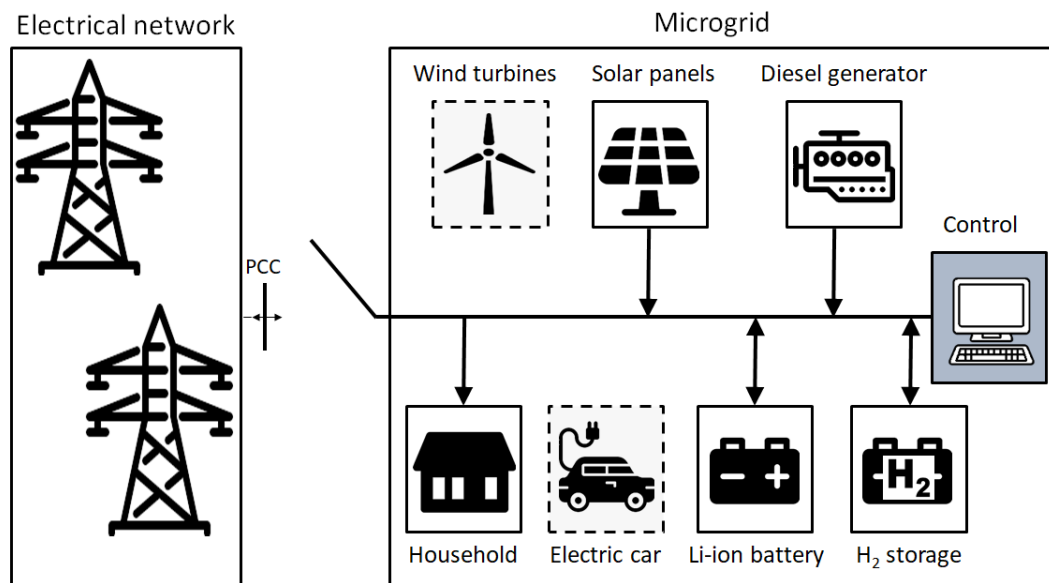


Figure 1. Schematic example of a microgrid.

2.2. Description of the Studied Microgrid

The main elements to be defined are the energy demand, the solar panel, the diesel generator and the energy storage system that consists of a Li-ion battery and a hydrogen tank. Both the physical characteristics of these elements in the studied microgrid, as well as the data used for fitting and testing the models, are discussed in later Section 4.1.

2.2.1. Demand

The load is modelled as a time series with hourly intervals that represent the average consumption of each hour, and it is assumed that historical data is available. Furthermore, this demand is not dependent on the behaviour of all the other elements of the microgrid. In case there is not enough generation to supply the microgrid demand, the non-served power will take the required positive value to ensure the demand balance equation. This non-served power will be severely penalised. However, in case there is an excess of RES generation, it will be assumed that such extra production can be curtailed without any cost (RES spillages).

2.2.2. Generation

The microgrid generation system consists of a solar panel and a diesel generator. Regarding the solar panel, the maximum PV generation profile can be represented by an hourly time series, and its variable operation cost can be neglected. On the other hand, the diesel generator can provide backup energy in case of a lack of RES production. This diesel generator can be dispatched by the control system that establishes the power to be produced in a deterministic manner, i.e., possible failures of the equipment is out of the scope of this paper. Two different operation modes will be considered. In the first one, the output power is allowed to take any continuous value between the minimum and the maximum power. In the second one, only discrete values are possible. For simplicity, the input-output cost curve is modelled as a quadratic function where the independent term, i.e., the no-load cost, is only incurred when the generator is committed. Start-up and shutdown costs are neglected, and further details can be seen in Section 4.

2.2.3. Storage

With respect to the storage, a Li-ion battery is used as short-term storage supporting the microgrid in periods when RES cannot supply the demand. It is characterised by the maximum capacity R_t^B

[kWh], the maximum power P_t^B [kW] for charge and discharge and the charge and discharge efficiencies μ_t^B, ζ_t^B . It is important to highlight, under the proposed RL approach, that it would be easy to take into account a more detailed model of the battery to capture the non-linear relationship among the state of charge (SoC), the maximum power, the efficiencies and other characteristics. However, as the classical optimisation model used to compare the results cannot include all those non-linear relationships, a simple model has been preferred. The other storage device, i.e., a hydrogen system, is used as long-term storage, supporting large periods of high demand and low RES. The characterisation of the hydrogen storage is analogous to the Li-ion battery: maximum capacity R_t^H [kWh], maximum power P_t^H [kW] for charge and discharge and the charging and discharging efficiencies μ_t^H and ζ_t^B .

3. Reinforcement Learning

RL belongs to the area of Machine Learning benefiting from ideas of optimal control of incompletely-known Markov decision process, and it has been applied to many fields [14]. An RL problem is modelled as an agent and an environment that interact with each other as in Figure 2. The agent makes decisions following a policy, exploring the environment dynamics and obtaining rewards. The agent updates its policy π guided by these rewards r_t over a discrete space of time divided in time steps t , improving the cumulative reward.

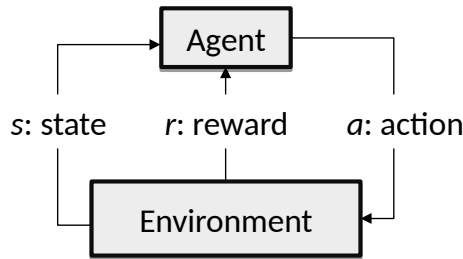


Figure 2. General interactions in Reinforcement Learning (RL).

In the RL context, a state space S and an action space A has to be defined. Given an initial state s_0 , a policy $\pi \in \Pi$ performs a trajectory $\tau = (s_0, a_0, s_1, a_1, \dots, s_n)$ where s_n is a terminal state. A reward function $R(s_t, a_t, s_{t+1})$ and a return function $G(\tau)$ have to be defined also according to the RL problem. The optimal policy π^* is the one that maximises the expected return. An intrinsic feature of an RL algorithm is the policy-based and the value-based characteristic, which affects the method of computing the next action and the learning behaviour. Some methods have both characteristics, but the proposed method in this paper has only the value-based characteristic.

3.1. Deep Reinforcement Learning (DRL)

Due to the curse of dimensionality, approximation functions are commonly used in more complex problems to approximate the policy, and given the significant impact of Deep Learning, a collection of RL algorithms using Artificial Neural Networks (ANN) has emerged, [15–17].

In this paper, a Deep Learning version of Q-Learning [18] called Deep Q-Network (DQN) is used [19], with a configuration and architecture similar to [5]. This method uses an ANN to approximate its value function, called Q-function. This method is model-free, i.e., the model does not use experiences to create an internal model to generate synthetic experiences and learn with, unlike model-based such as Dyna-Q [20] or Monte Carlo Tree Search (MCTS) [21]. Another characteristic of some of these algorithms is the capacity of work with both continuous and discrete action spaces, but DQN works only with discrete action spaces, hence, environments with continuous action spaces have to be discretised to be able to apply this algorithm in this kind of environments.

Deep Q-Network (DQN)

One of the most known algorithms in RL is Q-Learning. Every method in RL needs a policy function to know what action the agent has to take to perform the best trajectory and earn the best return from a state. In Q-Learning, the policy function is computed using the $Q(\cdot)$ function

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a'), \quad (1)$$

and with this function, the algorithm could perform the best believed action:

$$\text{best } a = \underset{a}{\operatorname{argmax}} Q(s, a). \quad (2)$$

The Q-function is updated as follows:

$$Q(s, a) = Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)], \quad (3)$$

with convergence to the optimal $Q^*(s, a)$. s' and a' are the next state and the action to be performed in s' respectively.

The curse of dimensionality phenomena makes the Q-Learning algorithm unsuitable to solve high dimensional problems, and approximations of the $Q(\cdot)$ function have to be introduced to deal with this drawback [14]. Since ANNs are universal approximators, DQN can be implemented to model the $Q_\theta(\cdot)$ function, being θ the parameters of the ANN. The updating process of the Q-values has to be computed to update each parameter θ of the ANN. Additionally, DQN uses a replay memory to help to efficiently update the parameters and get a good trade-off between exploration and exploitation [22].

3.2. Application of RL in the Microgrid

3.2.1. Action Definition

At every time period t , the RL algorithm must decide how much power should be charged or discharged at the Li-ion battery and hydrogen storage, and the power that should be produced by the diesel generator. Therefore, the agent action at time t is defined as:

$$a_t = (P_t^{h2}, P_t^d). \quad (4)$$

3.2.2. State Definition

The microgrid state can be defined as a tuple of k temporal slices. The idea is not only to rely on the instantaneous information of the microgrid at time step t , but also to consider the information recorded from the most recent stages. Therefore, a window-based state s_t is modelled for each time step as in [13]:

$$s_t = (\text{slice}_{t-k}, \text{slice}_{t-k+1}, \dots, \text{slice}_{t-1}, \text{slice}_t). \quad (5)$$

Regarding the variables to be monitored, in this paper it is proposed that each temporal slice is defined by the next variables:

$$\text{slice}_t = (P_{t-1}^{pv}, D_{t-1}, S_t^b, S_t^{h2}). \quad (6)$$

Notice that the state of charge (SoC) corresponds to the value of energy stored at the end of each hour, and that the generated or consumed power represents its average value during each hourly period.

In order to compute the SoC of both storage devices, the following energy balance equations must be taken into account:

$$S_t^b = \begin{cases} S_{t-1}^b - |P_t^b| / \zeta^b & \Leftrightarrow P_t^b \geq 0 \\ S_{t-1}^b + |P_t^b| \eta^b & \Leftrightarrow P_t^b < 0 \end{cases}, \quad (7)$$

$$S_t^{h2} = \begin{cases} S_{t-1}^{h2} - |P_t^{h2}| / \zeta^{h2} & \Leftrightarrow P_t^{h2} \geq 0 \\ S_{t-1}^{h2} + |P_t^{h2}| / \eta^{h2} & \Leftrightarrow P_t^{h2} < 0 \end{cases}. \quad (8)$$

Notice that both storage devices must be operated within their feasible limits, and this is ensured by a continuous monitoring at each decision step.

3.2.3. Reward Definition

The reward function, expressed in €, is directly related to the generation cost of the diesel generator, plus the possible penalty of the non-served power:

$$r_t = R(s_t, a_t, s_{t+1}) = -C^d(P_t^d) - c_{\text{pns}} P_t^{\text{pns}}, \quad (9)$$

where

$$C^d(x) = \delta_2 x^2 + \delta_1 x + \delta_0 \quad (10)$$

is the diesel cost function [€] with quadratic coefficients δ_2 , δ_1 and δ_0 . P_t^b , P_t^{pns} and P_t^{curt} are calculated using the demand balance-constraint equation given by

$$(D_t - P_t^{\text{pns}}) = (P_t^{\text{pv}} - P_t^{\text{curt}}) + P_t^b + P_t^{h2} + P_t^d, \quad (11)$$

prioritising the usage of the battery to reduce the not-supplied power and the curtailment.

4. Case Study

This section describes the case study used in this paper, as well as the different configurations of the proposed Deep Reinforcement Learning (DRL) that have been used to perform the comparative analysis.

4.1. Description of the Data

The data set consists of three consecutive years with hourly values. Therefore, $|T| = 3 \times 8760 = 26,280$. The input data are the solar panel maximum generation profile, the hourly load, and all the technical characteristics of the other components. All these parameters are explained in detail hereafter and summarised in Table 1.

Table 1. Microgrid parameters.

| Component | Parameter | Value |
|----------------|---------------------------------|--------|
| Solar Panel | P_{\max}^{pv} [kW] | 6 |
| Load | D_{\max} [kW] | 2.1 |
| Diesel | P_{\max}^d [kW] | 1.0 |
| | δ_2 [€/kW ²] | 0.31 |
| | δ_1 [€/kW] | 0.108 |
| | δ_0 [€] | 0.0157 |
| Li-ion battery | P_{\max}^b [kW] | 2.9 |
| | S_0^b [kWh] | 0 |
| | S_{\max}^b [kWh] | 2.9 |
| | ζ^b | 0.95 |
| | η^b | 0.95 |

Table 1. Cont.

| Component | Parameter | Value |
|----------------------------|----------------------|-------|
| H_2 storage | P_{max}^{h2} [kW] | 1.0 |
| | S_0^{h2} [kWh] | 100 |
| | S_{max}^{h2} [kWh] | 200 |
| | ζ^{h2} | 0.65 |
| | η^{h2} | 0.65 |
| cost of not supplied power | c_{pns} [€/kWh] | 1 |

4.1.1. Photovoltaic Panels

The solar generation profile is given by the irradiation data used from [5] that can be obtained from the the Deep Reinforcement learning framework (DeeR) python library [23]. This data was collected from a Belgium location and the factor of the aggregated irradiation by month is 1:5 between the lowest and the highest monthly available solar generation. The optimal sizing of the microgrid is out of the scope of this paper. Therefore, the number of panels has been chosen just by dividing the annual demand by the total irradiation taking into account a standard efficiency for the PV panels. As a result, the PV system dimension for the case study is 30 m², with a maximum installed power of 6 kW, which is half of the size proposed by [24]. Furthermore, some websites as [25] discuss that a 6 kW PV system is a commonly used figure in the U.S.

4.1.2. Load

The hourly data for the load profile has also been obtained from [23]. This load profile represents the typical residential consumption of a consumer with an average daily demand of 18.33 kWh. In case the microgrid is unable to supply all the load, it is assumed a cost 1€ per not supplied kWh.

4.1.3. Energy Storage Elements

The Li-ion battery charge and discharge efficiency rates are 0.95 for both processes. For the hydrogen (H_2) electrolyser (charge) and fuel cell (discharge) the efficiency rates are 0.65 for both cases, [24]. The battery capacity and hydrogen storage size are 2.9 kWh and 200 kWh. The battery model has the characteristics of an LG Chem RESU3.3 used for households, with a maximum power of 3.0 kW but clipped to 2.9 kW for the microgrid (due to the hourly step). The hydrogen fuel cell maximum power is 1.0 kW based on a Horizon 1000W PEM fuel cell, and for symmetry 1.0 kW for the electrolyser.

4.1.4. Non-Renewable Generation

The proposed microgrid has a diesel generator with a nominal rate of 1.0 kW. The cost curve of this diesel generator has been adjusted to a quadratic-curve (see Equation (10)), adapting the parameters of the IEEE 30 bus system generators used in [26]. Table 1 shows the obtained coefficients of the obtained cost polynomial.

4.2. Optimisation-Based Model (MIQP) Used as a Reference Model and Benchmark

In the hypothetical case where the microgrid's hourly demand and solar power for the entire time horizon were known, it would be possible to formulate a deterministic optimisation problem in order to obtain the most favourable scheduling of all the elements of the microgrid. The value of the objective function in this setting would represent the lower bound (if the objective is to minimise the operation cost), and it could serve as a benchmark value to compare the results obtained by alternative methods. The decision variables of this optimisation problem are the charge and discharge power

of the batteries, the possible generation of the diesel group, the not supplied power and the possible spillages. This benchmark model was implemented in GAMS [27], where the detailed equations were omitted here for the sake of simplicity, but added in Appendix A. The structure of the resulting optimisation problem is the following one:

$$\begin{aligned} \min_x \quad & c(x) \\ \text{subject to} \quad & g(x) \leq 0 \end{aligned} \quad , \quad (12)$$

where the objective function $c(x)$ is the sum of the diesel generation cost along the entire time horizon plus the penalty term related to the possible non-supplied energy. The cost of the diesel generator is assumed to be a quadratic function, where the independent term is only incurred when the generator is on. This fact requires the use of unit-commitment binary variables.

The constraints $g(x)$ are the energy balance at the microgrid, and at the storage devices taking into account their charging and discharging efficiencies, their maximum storage levels and the maximum rated power.

As the objective is a quadratic function and some binary variables are needed, the resulting model is a Mixed Integer Quadratic Programming model (MIQP).

4.3. Naive Strategy

A Naive algorithm has also been implemented to mimic the results that a simple strategy could achieve. The insight of this method is to charge the batteries when there is a surplus of energy, and discharge them otherwise. The pseudocode of this strategy is shown in Algorithm 1, where PVGEN is the PV power [kW] and the LOAD is the household consumed power [kW].

Algorithm 1 Naive algorithm

```

1: if PVGEN > LOAD then
2:   extra = PVGEN - LOAD
3:   if extra > batt capacity then
4:     batt charge = min(batt capacity, Pb)
5:   else
6:     batt charge = min(extra, Pb)
7:   end if
8:   extra = extra - batt charge
9:   if extra > hyd capacity then
10:    hyd charge = min(hyd capacity, Ph2)
11:   else
12:    hyd charge = min(extra, Ph2)
13:   end if
14:   extra = extra - hyd charge
15: else
16:   lack = LOAD - PVGEN
17:   if lack > batt capacity then
18:     batt discharge = min(batt capacity, Pb)
19:   else
20:     batt discharge = min(lack, Pb)
21:   end if
22:   lack = lack - batt discharge
23:   if lack > hyd capacity then
24:     hyd discharge = min(hyd capacity, Ph2)
25:   else
26:     hyd discharge = min(lack, Ph2)
27:   end if
28:   lack = lack - hyd discharge
29:   if lack > 0 then
30:     diesel = min(max diesel, lack)
31:   end if
32: end if

```

It is assumed that the Naive algorithm could be implemented in a microcontroller making decisions on a continuous manner based on the instantaneous status of the microgrid. In this sense, this differs from the DRL approach of this paper that would require at least the information of the last hour in case the size of the time window was 1 h.

4.4. DQN Configuration

This paper uses a DQN due to its implementation simplicity and powerful performance [19]. It consists of three modules:

- (1) A Convolutional Neural Network (CNN) that it is shown in Figure 3.
- (2) A replay memory to store the experiences and train the CNN. The implemented memory is called Experience Replay Memory [28], being the easiest to implement.
- (3) A module to process the agent observations and combine them with internal agent data (the last state), to make the next state and store it in the memory.

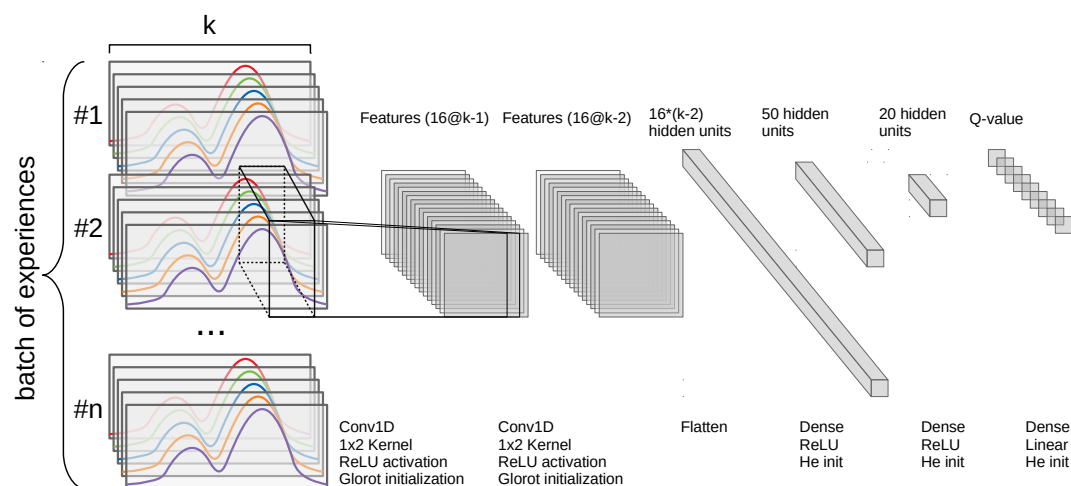


Figure 3. Convolutional Neural Network architecture used in the proposed Deep Q-Network (DQN).

5. Results

The CNN architecture proposed is similar but not equal to the one proposed in [5]. It takes a whole tensor as the input of the first layer instead of scatter time-series input in 1D-convolution layers and being merged with the remain inputs after the convolution computation. This configuration brings more simplicity and scalability to the model without losing performance in the results despite the information redundancy. This CNN has two 1-dimension convolution layer. The initialisation procedures of these layers are the ones from Glorot [29] for Convolutional layers and He [30] for Dense layers. All the parameter values used are shown in Table 2.

Table 2. DQN parameters.

| Parameter | Value |
|------------------------------|----------------------------------|
| Batch size | 20 |
| Memory size | 10,000 |
| Optimiser | Nadam [31] |
| Error measure | MSE |
| Exploration function | $f(s) = 0.1 + 0.9e^{-s*10^{-6}}$ |
| Early stop scope | 200 |
| Discount factor (γ) | 0.99 |

During the training phase, to deal with the lack of data, a regularisation technique called early stop is applied. In this technique, the training set is split into a smaller training set and the development

set, and after a fixed number of training steps, the model is evaluated with the development set saving the best snapshot of the model during the training.

This paper carries out the analysis of the window size of the state throughout different configurations of the parameter k in Equation (5). The window size ranges from $k = 3$ to $k = 24$ in steps of 3 h. Following the state definition of Equation (5), some parameters in $t < 0$ are required (state s_0 contains some temporal slices in $t < 0$), and these are set to zero.

This section presents the results obtained with the DQN method applied to the microgrid described previously. The DQN has been implemented in Python 3.7.7 using Tensorflow 2.1.0. The MIQP model has been coded in GAMS using the solver CPLEX 12.9.0.0. The computer system was an Intel Core i7-8550U (1.80 GHz–4.00 GHz) with 16 GB RAM running under Ubuntu 18.04 LTS x64.

The results gather several options for the definition of the state of the microgrid varying the window size parameter k in Equation (5). As stated before, the motivation of this analysis is to find how much recent information should be included in the state definition in order to obtain the best performance of the microgrid. To put the results obtained with the DQN in perspective, other models have been included in this analysis: the perfect-information MIQP model, the Naive strategy and a Random Policy. The first one represents a lower bound of the operation cost that can be achieved with such microgrid. The third one can be seen as an upper bound, where the average value of ten samples (runs of the model) are included in the assessment. Based on the historical three-year data, the return $G(\tau)$ of the trajectory τ followed by the policy of the proposed DQN is used to measure its goodness. The return match the cost of the microgrid in € and the trajectory τ is the operation of the microgrid. Note that the first two years of available data are used to train the DQN model, separated in the training set of the first year and the development set of the second year, whereas the last year is used for testing. Concerning the reference provided by the MIQP, the model operates the microgrid without any discretisation of the decision variables. The same applies for the Naive and the Random strategies.

The MIQP model generates the lower bound given by a three-year accumulated cost of 2677.43 € with a relative GAP of 6.06 as GAMS defines $(BestFeasible - BestPossible) / BestFeasible$, taking 24 h to find this solution. The accumulated cost for every year is shown in Table 3. The Naive strategy achieves an expenses of 11,138.60 €, and the Random strategy 14,066.59 €, 316.02% and 425.38% more than MIQP, respectively. Notice that the Naive strategy is not much better than the Random strategy, and this fact calls for smarter policies to operate the microgrid, as the one proposed in this paper.

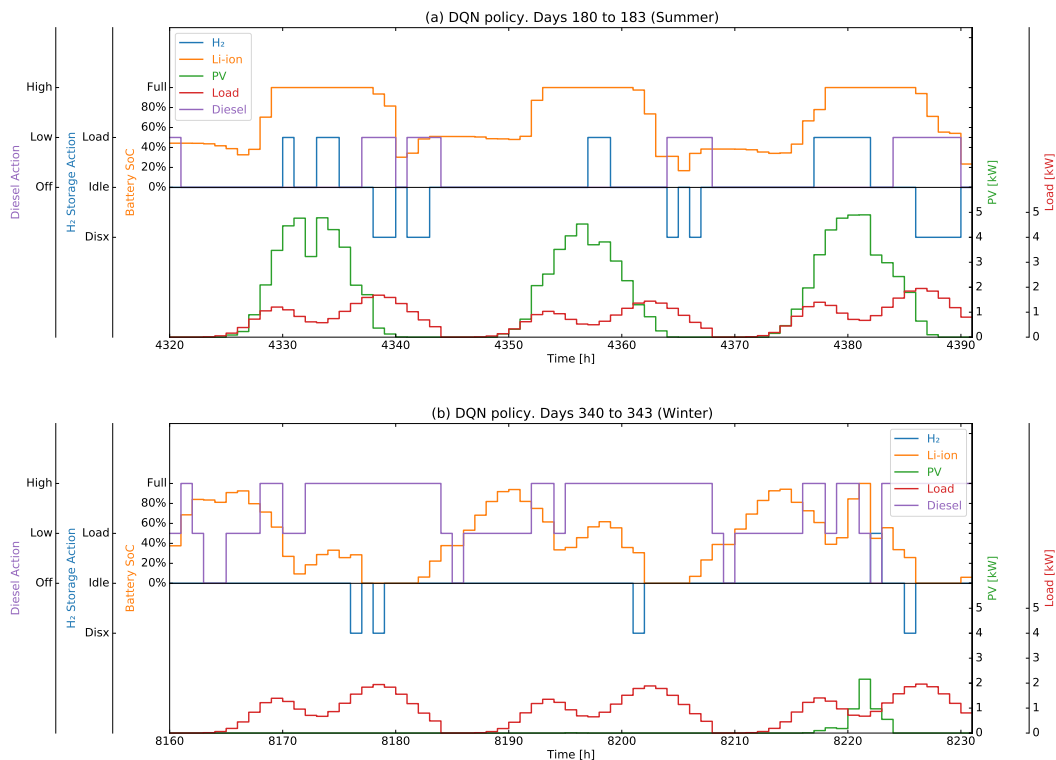
Table 3 shows the accumulated cost obtained after three years of operation of the microgrid using the policy learned, for the different configurations of the proposed DQN. These results are significantly better than the other models, achieving results between 3653.59 € for the best case with a window size of 9, and 462,722 € for the worst-case with a window size of 21. Even the worst-case is much better than a Naive or Random strategies. Relative to the results of the MIQP model, the worst-case performance is two times worse than the best case. These results show that the choice of the window size parameter has significant consequences in the performance of the DQN. Larger window size involves learning difficulties in the policy for the DQN.

Another observation is the comparative between the best case DQN and the MIQP results. The DQN achieves a good performance, being a method easy to implement and with near-optimal results.

Analysing the agent behaviour, Figure 4 shows an example of the hourly microgrid operation carried out by the proposed RL method. It consists of six days of the first year (three consecutive days of winter and three of summer).

Table 3. Accumulated cost of each algorithm [€].

| Cost [€] | Year 1 | Year 2 | Year 3 | Total | Relative $(X - BEST)/BEST \cdot 100$ |
|---------------|---------|---------|---------|-----------|--------------------------------------|
| MIQP (GAP 6%) | 967.34 | 864.05 | 846.04 | 2677.43 | 0.00% |
| RL k = 3 | 1305.94 | 1126.49 | 1239.35 | 3671.77 | 37.14% |
| RL k = 6 | 1355.80 | 1140.92 | 1248.61 | 3745.33 | 39.89% |
| RL k = 9 | 1299.56 | 1123.53 | 1230.50 | 3653.59 | 36.46% |
| RL k = 12 | 1389.60 | 1198.63 | 1308.11 | 3896.33 | 45.52% |
| RL k = 15 | 1348.61 | 1174.02 | 1268.02 | 3790.64 | 41.58% |
| RL k = 18 | 1503.69 | 1307.67 | 1443.58 | 4254.94 | 58.92% |
| RL k = 21 | 1634.33 | 1441.21 | 1551.68 | 4627.22 | 72.82% |
| RL k = 24 | 1515.46 | 1320.44 | 1439.12 | 4275.02 | 59.67% |
| Naive | 3778.74 | 3681.04 | 3678.82 | 11,138.60 | 316.02% |
| Random | 4816.64 | 4554.78 | 4695.17 | 14,066.59 | 425.38% |

**Figure 4.** DQN operation of the microgrid, for summer (a) and winter (b). Hourly steps in a 3-day window.

Since the DQN only works with discrete action spaces, the continuous action space defined in the MDP model of the microgrid in Equation (4) has to be discretised. Each hour t , the action a_t consists of two variables (P_t^d and P_t^{h2}) corresponding to the power generated by the diesel generator and the power generated (consumed if negative) by the hydrogen fuel cell respectively. The required discretisation of the diesel output power P_t^d leads to the following cases:

- $P_t^d = P_{max}^d$, i.e., the agent configures the diesel generator to dispatch the maximum power (1 kW in the example, and “High” label in Figure 4).
- $P_t^d = P_{max}^d/2$, i.e., the agent configures the diesel generator to dispatch half of the maximum power it can (0.5 kW in the example, and “Low” label in Figure 4).
- $P_t^d = 0$, i.e., the agent configures the diesel generator in off. (0 kW in the example, and “Off” label in Figure 4).

The agent considers also three different configurations for the values that P_t^{h2} can take:

- $P_t^{h2} = -P_{max}^{h2}$, i.e., the agent configures the hydrogen electrolyser to charge at maximum rate from the system (load at 1 kW in the example, and “Load” label in Figure 4).
- $P_t^{h2} = 0$, i.e., the agent configures the hydrogen components in an idle state (load/discharge at 0 kW, and “Idle” label in Figure 4).
- $P_t^{h2} = P_{max}^{h2}$, i.e., the agent configures the hydrogen fuel cell to inject the maximum power into the system (discharge at 1 kW in the example, and “Disx” label in Figure 4).

To summarise, in each hour the agent can take nine different configurations. The battery SoC takes values between 0 kWh (0% in the figure) and 2.9 kWh (Full in the figure that correspond to 100% of the maximum SoC).

According to Figure 4, there are no significant differences between the summer and winter load profiles. In contrast, the available solar irradiation makes the difference between these opposite seasons, being the main responsible for the change in the operation policy decided automatically by the RL method.

There is a clear daily pattern operation of the Li-ion battery, but it differs with the season. During summer the daily energy surplus provided by the solar panels is used to fill this battery during daylight hours, being systematically discharged during the rest of the hours. However, the lack of solar irradiation in winter is compensated by the continuous generation of the diesel to cover the demand in the main hours, using the diesel surplus of the off-peak hours to charge the battery for its usage during non-sunlight hours. On the contrary, during summer the use of the diesel is limited to some hours where there is not enough solar irradiation. Concerning the hydrogen fuel cell, it is used in a similar diesel pattern, to cover the demand when there is not enough solar irradiation.

In order to illustrate the behaviour of the model, a numerical example is presented hereafter. Assuming that the state s_t considers a tensor of shape $(k, 4)$ with $k = 9$, from the output results shown in Figure 4a, the detailed values that correspond to hour $t = 4380$ are the next ones:

$$\begin{aligned}
 s_t &= \left(\text{slice}_{t-k}, \text{slice}_{t-k+1}, \dots, \text{slice}_{t-1}, \text{slice}_t \right) \\
 &= \begin{pmatrix} P_{t-k-1}^{pv} & P_{t-k}^{pv} & \dots & P_{t-2}^{pv} & P_{t-1}^{pv} \\ D_{t-k-1} & D_{t-k} & \dots & D_{t-2} & D_{t-1} \\ S_{t-k}^b & S_{t-k+1}^b & \dots & S_{t-1}^b & S_t^b \\ S_{t-k}^{h2} & S_{t-k+1}^{h2} & \dots & S_{t-1}^{h2} & S_t^{h2} \end{pmatrix} \\
 &= \begin{pmatrix} 0.002, & 0.151, & 0.461, & 1.122, & 1.973, & 3.301, & 4.295, & 4.767, & 4.891 \\ 0.061, & 0.186, & 0.447, & 0.837, & 1.222, & 1.398, & 1.270, & 0.963, & 0.711 \\ 1.028, & 0.989, & 1.001, & 1.257, & 1.932, & 2.260, & 2.900, & 2.900, & 2.900 \\ 36.000, & 36.000, & 36.000, & 36.000, & 36.000, & 36.650, & 37.300, & 37.950, & 38.600 \end{pmatrix}.
 \end{aligned}$$

Then, the action taken by the DQN using Equation (2) is:

$$\begin{aligned}
 a_t &= \left(P_t^d, P_t^{h2} \right) = \\
 &\quad \left(0.000, -1.000 \right)
 \end{aligned}$$

As a consequence of this action, the new state in hour $t + 1 = 4381$ is reached, with its corresponding new slice:

$$\begin{aligned} slice_{t+1} &= (P_t^{pv}, D_t, S_{t+1}^b, S_{t+1}^{h2}) \\ &= (4.899, 0.672, 2.900, 39.250), \end{aligned}$$

in which S_{t+1}^b , that is calculated using the Equation (A3), takes the value 0.0 kW because it is full and the curtailment P_t^{curt} , that is calculated using the balance Equation (11), takes the value of 3.217 kW.

Another interesting result that can be highlighted is the management of the hydrogen storage along the year. One could expect that since there is less solar generation in winter, it would be better to store energy during summer in order to have it available when necessary. Figure 5 shows the hydrogen storage SoC for the whole 3-year period since this profile is barely appreciable in the 3-day window of Figure 4. It can be seen that the proposed DQN is capable of finding an optimal yearly pattern, charging the hydrogen tank in summer and discharging it in winter, and this optimal behaviour is obtained just with the most recent information at each step. A summary of the usage of the hydrogen storage is shown in Table 4.

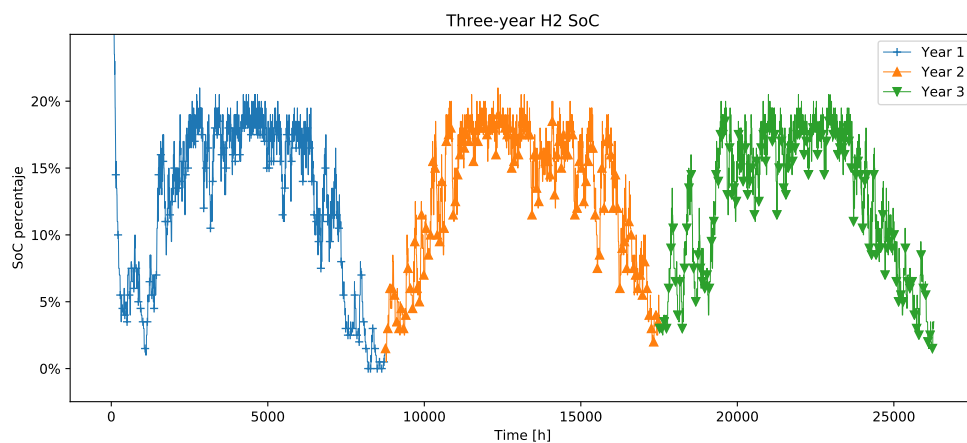


Figure 5. Hydrogen state of charge (SoC) percentage over three years.

Table 4. Hydrogen storage SoC fluctuation.

| Hydrogen Storage | Year 1 | Year 2 | Year 3 |
|------------------|------------|------------|------------|
| Charged | 546.00 kWh | 587.50 kWh | 578.50 kWh |
| Discharged | 594.50 kWh | 586.00 kWh | 578.00 kWh |

6. Conclusions

This paper presents the application of DRL techniques to manage the elements of an isolated microgrid. The advantage of the proposed approach is the ability of the algorithm to learn from its own experience, and this can be an interesting feature to foster the up-scaling of these type of solutions that do not require a tailor-made specific optimisation model adapted to the particularities of each microgrid. However, as the application of RL and DRL to power systems is an emerging research field, the selection of the most appropriate ANN architecture, or the definition of what variables should be included in the so-called system state, are still open questions. With respect to the first one, the proposed CNN architecture provides simplicity and a good performance. Regarding the second question, this paper provides a sensible set of variables to configure the system state. In addition,

the effect of increasing the time window of such state has been analysed, and the numerical results show that 9 h is the optimal size of such time window for the studied microgrid. In order to measure the quality of the DRL results, this paper presents a traditional Mixed Integer Quadratic Programming optimisation model to compute the theoretical best operation possible, i.e., the hypothetical case of having complete information for the whole time period. Using such model as benchmark, whereas a Naive algorithm similar to the ones used in practical implementation results in an operation 316.02% worse, the developed DRL model reaches a value of 36.46%. Future work will be oriented to the implementation of Transfer Learning and Robust Learning to address multi-agent microgrid problems.

Author Contributions: Conceptualisation, J.G.-G. and M.A.S.-B.; methodology, D.D.-B., J.G.-G. and M.A.S.-B.; software, D.D.-B.; validation, D.D.-B., J.G.-G., M.A.S.-B. and E.F.S.-Ú.; formal analysis, D.D.-B., J.G.-G. and M.A.S.-B.; data curation, D.D.-B. and E.F.S.-Ú.; writing—original draft preparation, D.D.-B.; writing—review and editing, D.D.-B., J.G.-G., M.A.S.-B. and E.F.S.-Ú.; visualisation, D.D.-B.; supervision, J.G.-G. and M.A.S.-B. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been funded by the Strategic Research Grants program of Comillas University, and by “Programa microrredes inteligentes, Comunidad de Madrid (P2018/EMT4366)”.

Acknowledgments: The authors want to express their gratitude to Francisco Martín Martínez and Miguel M. Martín-Lopo for their valuable help during the development of this research.

Conflicts of Interest: The authors declare no conflict of interest.

Notation

To facilitate the mathematical representation of the elements of the microgrid, the following nomenclature is used throughout the paper.

Sets and indexes:

$t \in T$ time periods from 1 to $|T|$

Parameters:

| | |
|----------------|--|
| P_t^{pv} | Maximum available PV power generation at time period t , [kW] |
| P_{max}^{pv} | Nominal rate of the solar panel, [kW] |
| D_t | Load of the microgrid at time period t , [kW] |
| D_{max} | Maximum power the microgrid can consume, [kW] |
| ζ^b | Discharge efficiency of the Li-ion battery |
| η^b | Charge efficiency of the Li-ion battery |
| ζ^{h2} | Discharge efficiency constants of the hydrogen storage |
| η^{h2} | Charge efficiency hydrogen storage |
| δ_2 | Quadratic term of the diesel generator cost function, [€/kW ²] |
| δ_1 | Linear term of the diesel generator cost function, [€/kW] |
| δ_0 | No-load term of the diesel generator cost function when committed, [€] |
| c_{pns} | Cost of not supplied energy, [€/kW] |
| P_{max}^d | Maximum output power of the diesel generator, [kW] |
| P_{max}^b | Maximum power the Li-ion battery can dispatch, [kW] |
| S_{max}^b | Maximum SoC of the battery, [kWh] |
| S_0^b | Initial SoC of the battery, [kWh] |
| P_{max}^{h2} | Maximum power the hydrogen fuel cell can dispatch, [kW] |
| S_{max}^{h2} | Maximum SoC of the hydrogen storage, [kWh] |
| S_0^{h2} | initial SoC of the hydrogen storage, [kWh] |

Variables:

| | |
|--------------|--|
| P_t^d | Output power of the diesel generator at time period t , [kW] |
| P_t^b | Generated (+) and consumed (-) power from the Li-ion battery at time period t , [kW] |
| P_t^{h2} | Generated (+) and consumed (-) power from the hydrogen storage at time period t , [kW] |
| S_t^b | State of Charge (SoC) of the Li-ion battery at time period t , [kWh] |
| S_t^{h2} | State of Charge (SoC) of the hydrogen storage at time period t , [kWh] |
| P_t^{pns} | Not-supplied power at time period t , [kW] |
| P_t^{curt} | Non-negative variable that represents the PV curtailment at time period t , [kW] |

Variables (only for MIQP):

| | |
|-------------|---|
| U_t^d | Unit commitment of diesel at t |
| P_t^{b+} | Consumed power from the Li-ion battery at time period t , [kW] |
| P_t^{b-} | Generated power from the Li-ion battery at time period t , [kW] |
| P_t^{h2+} | Hydrogen electrolyser consumption at t , [kW] |
| P_t^{h2-} | hydrogen fuel cell generation at t , [kW] |

Abbreviations

The following abbreviations are used in this manuscript:

| | |
|------|-------------------------------------|
| ANN | Artificial Neural Network |
| CNN | Convolutional Neural Network |
| DQN | Deep Q-Network |
| DRL | Deep Reinforcement Learning |
| GAMS | General Algebraic Model System |
| LTS | Long-Term Support |
| MCTS | Monte Carlo Tree Search |
| MIQP | Mixed Integer Quadratic Programming |
| MSE | Mean Squared Error |
| PCC | Point of Common Coupling |
| PV | Photovoltaic Panel |
| RES | Renewable energy sources |
| RL | Reinforcement Learning |
| SoC | State of Charge |
| pns | power not supplied |

Appendix A. Mixed Integer Quadratic Programming Formulation

$$\min \quad \sum_t [\delta_0 U_t^d + \delta_1 P_t^d + \delta_2 (P_t^d)^2 + c_{pns} P_t^{pns}] \quad (A1)$$

$$\text{subject to} \quad D_t - P_t^{pns} = (P_t^{pv} - P_t^{curt}) + (P_t^{b-} - P_t^{b+}) + (P_t^{h2-} - P_t^{h2+}) + P_t^d \quad \forall t \in T \quad (A2)$$

$$S_t^b = -\frac{P_t^{b-}}{\eta^b} + \zeta^b P_t^{b+} + S_{t-1}^b \quad \forall t \in T \quad (A3)$$

$$S_t^{h2} = -\frac{P_t^{h2-}}{\eta^{h2}} + \zeta^{h2} P_t^{h2+} + S_{t-1}^{h2} \quad \forall t \in T \quad (A4)$$

$$S_0^{h2} \leq S_t^{h2} \quad t = |T| \quad (A5)$$

$$P_t^d \leq U_t^d * \bar{P}^d \quad \forall t \in T \quad (A6)$$

$$U_t^d \in \{0, 1\} \quad \forall t \in T \quad (A7)$$

$$P_t^{pns}, P_t^d, P_t^{curt}, P_t^{b-}, P_t^{b+}, P_t^{h2-}, P_t^{h2+} \geq 0 \quad \forall t \in T \quad (A8)$$

$$P_t^d \leq P_{\max}^d \quad \forall t \in T \quad (A9)$$

$$P_t^{curt} \leq P_t^{pv} \quad \forall t \in T \quad (A10)$$

$$P_t^{b-}, P_t^{b+} \leq P_{\max}^b \quad \forall t \in T \quad (A11)$$

$$P_t^{h2-}, P_t^{h2+} \leq P_{\max}^{h2} \quad \forall t \in T, \quad (A12)$$

where Equation (A1) is the objective function, Equation (A2) is the demand balance constraint, Equations (A3) and (A4) model the dynamics of the battery and the hydrogen storage, respectively, Equation (A5) establishes a minimum hydrogen storage level in the last hour, and Equation (A6) indicates that unit commitment of the diesel generator is a binary variable. The remaining equations impose the upper and lower bounds to the decision variables.

References

1. Ali, A.; Li, W.; Hussain, R.; He, X.; Williams, B.; Memon, A. Overview of Current Microgrid Policies, Incentives and Barriers in the European Union, United States and China. *Sustainability* **2017**, *9*, 1146, doi:10.3390/su9071146.
2. Gupta, R.A.; Gupta, N.K. A robust optimization based approach for microgrid operation in deregulated environment. *Energy Convers. Manag.* **2015**, *93*, 121–131, doi:10.1016/j.enconman.2015.01.008.
3. Reddy, S.; Vuddanti, S.; Jung, C.M. Review of stochastic optimization methods for smart grid. *Front. Energy* **2017**, *11*, 197–209, doi:10.1007/s11708-017-0457-7.
4. Kim, R.K.; Glick, M.B.; Olson, K.R.; Kim, Y.S. MILP-PSO Combined Optimization Algorithm for an Islanded Microgrid Scheduling with Detailed Battery ESS Efficiency Model and Policy Considerations. *Energies* **2020**, *13*, 1898, doi:10.3390/en13081898.
5. François-Lavet, V.; Taralla, D.; Ernst, D.; Fonteneau, R. Deep Reinforcement Learning Solutions for Energy Microgrids Management. In Proceedings of the European Workshop on Reinforcement Learning (EWRL 2016), Barcelona, Spain, 3–4 December 2016.
6. Mbuwir, B.V.; Ruelens, F.; Spiessens, F.; Deconinck, G. Battery Energy Management in a Microgrid Using Batch Reinforcement Learning. *Energies* **2017**, *10*, 1–19.
7. Kim, B.; Zhang, Y.; van der Schaar, M.; Lee, J. Dynamic Pricing and Energy Consumption Scheduling with Reinforcement Learning. *IEEE Trans. Smart Grid* **2016**, *7*, 2187–2198, doi:10.1109/TSG.2015.2495145.
8. Qiu, X.; Nguyen, T.A.; Crow, M.L. Heterogeneous Energy Storage Optimization for Microgrids. *IEEE Trans. Smart Grid* **2016**, *7*, 1453–1461, doi:10.1109/TSG.2015.2461134.

9. Glavic, M.; Fonteneau, R.; Ernst, D. Reinforcement Learning for Electric Power System Decision and Control: Past Considerations and Perspectives. *IFAC-PapersOnLine* **2017**, *50*, 6918–6927, doi:10.1016/j.ifacol.2017.08.1217.
10. Glavic, M. (Deep) Reinforcement learning for electric power system control and related problems: A short review and perspectives. *Annu. Rev. Control* **2019**, *48*, 22–35, doi:10.1016/j.arcontrol.2019.09.008.
11. Yang, T.; Zhao, L.; Li, W.; Zomaya, A.Y. Reinforcement learning in sustainable energy and electric systems: A survey. *Annu. Rev. Control* **2020**, doi:10.1016/j.arcontrol.2020.03.001.
12. Ji, Y.; Wang, J.; Xu, J.; Fang, X.; Zhang, H. Real-Time Energy Management of a Microgrid Using Deep Reinforcement Learning. *Energies* **2019**, *12*, 2291, doi:10.3390/en12122291.
13. Francois-Lavet, V.; Rabusseau, G.; Pineau, J.; Ernst, D.; Fonteneau, R. On Overfitting and Asymptotic Bias in Batch Reinforcement Learning with Partial Observability. *J. Artif. Intell. Res.* **2019**, *65*, 1–30, doi:10.1613/jair.1.11478.
14. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; The MIT Press: Cambridge, England, 2018.
15. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. A Brief Survey of Deep Reinforcement Learning. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38, doi:10.1109/MSP.2017.2743240.
16. Li, Y. Deep Reinforcement Learning: An Overview. *arXiv* **2017**, arXiv: 1701.07274.
17. Francois-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M.G.; Pineau, J. An Introduction to Deep Reinforcement Learning. *Found. Trends® Mach. Learn.* **2018**, *11*, 219–354, doi:10.1561/22000000071.
18. Watkins, C.J.C.H.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292, doi:10.1007/BF00992698.
19. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602.
20. Sutton, R.S. Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming. In Proceedings of the Seventh International Conference on Machine Learning, Morgan Kaufmann, Austin, TX, USA, 21–23 June 1990; pp. 216–224.
21. Chaslot, G.; Bakkes, S.; Szita, I.; Spronck, P. Monte-Carlo Tree Search: A New Framework for Game AI. In Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference, Palo Alto, CA, USA, 22–24 October 2008; Michael Mateas and Chris Darken; AAAI Press: Menlo Park, CA, USA, 2008; pp. 216–217.
22. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533, doi:10.1038/nature14236.
23. François-Lavet, V.; Taralla, D. DeeR. 2016. Available online: <http://deer.readthedocs.io> (accessed on 16 April 2020).
24. François-Lavet, V.; Gemine, Q.; Ernst, D.; Fonteneau, R. *Towards the Minimization of the Levelized Energy Costs of Microgrids using Both Long-Term and Short-Term Storage Devices*; CRC Press: Boca Raton, Florida 2016.
25. EnergySage. EnergySage: 6 kW Solar System. 2019. Available online: <http://news.energysage.com/6kw-solar-system-compare-prices-installers> (accessed on 16 April 2020).
26. Jasmin, E.A.; Imthias Ahamed, T.P.; Jagathy Raj, V.P. Reinforcement Learning approaches to Economic Dispatch problem. *Int. J. Electr. Power Energy Syst.* **2011**, *33*, 836–845, doi:10.1016/j.jepes.2010.12.008.
27. Corporation, G.D. General Algebraic Modeling System (GAMS) Release 24.2.1. 2013. Available online: <http://www.gams.com/> (accessed on 16 April 2020).
28. Lin, L.J. *Reinforcement Learning for Robots Using Neural Networks*; Technical Report CMU-CS-93-103; School of Computer Science, Carnegie Mellon University: Pittsburgh, PA, USA, 1993.
29. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010; Teh, Y.W., Titterton, M., Eds.; Proceedings of the Machine Learning Research (PMLR); 2010; Volume 9, pp. 249–256. Available online: <http://proceedings.mlr.press/v9/glorot10a.html> (accessed on 16 April 2020).

30. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
31. Dozat, T. Incorporating Nesterov Momentum into Adam. In Proceedings of the ICLR 2016 Workshop, San Juan, Puerto Rico, 2–4 May 2016.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).