Machine Learning applied to the operation of fully renewable energy systems

Ruben Chaer
Facultad de Ingeniería
Universidad de la República and
Electricity Market Administration
Montevideo, Uruguay
rchaer@simsee.org

Ignacio Ramírez
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
nacho@fing.edu.uy

Gonzalo Casaravilla
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
gcp@fing.edu.uy

***Abstract*—This work presents a novel learning algorithm for the operation policy of power systems trying to minimize the cost of fulfilling the energy demand. The algorithm improves upon the classical reinforcement learning methods by controlling the sampling variance in the estimation of the future cost spatial differences, together with parameter regularization and dynamic exploring techniques. The proposed strategy was applied to a case of what could be the power system of Uruguay by 2050 based strongly in hydro, wind and solar energies, including three lakes, four groups of battery banks, and the basin runoff of the two main Uruguayan rivers. The generation in the year 2022 in Uruguay was 43% hydraulic, 40% wind plus solar, 7% biomass and 10% based on fossil fuels. This composition prints a very relevant stochastic component that makes it difficult to apply machine learning techniques without the kind of algorihms proposed in this work.**

Keywords—Approximate Stochastic Dynamic Programming, Reinforcement Machine Learning, Renewable Energies.

I. Introduction

The optimal operation of electrical energy systems with storage capacity falls within the category of dynamic stochastic programming problems. One of the forerunners in proposing a solution to the problem was Richard Bellman [1]. There it was shown that it is possible to obtain an Optimal Operation Policy based on a recursive estimation of what is known as the State Value function, Future Cost (FC) function or Cost-To-Go function, a method known today as the *Bellman Recursion* (BR).

The optimal operation of hydrothermal systems is a challenging task, especially in Latin American countries, characterized by a high hydroelectric component. Programming the energy dispatch involves choosing which resources will be used to guarantee the supply of the system load in the following hours, days, months and years, while minimizing the overall cost and complying with safety and quality requirements. In the presence of energy storage elements (e.g., hydroelectric lakes and batteries), the problem becomes a Stochastic Dynamic Programming (SDP) one.

In [1], it was already observed that the Bellman Recursion algorithm quickly becomes impractical when the combined dimension of the state and random variable spaces increases: this is known as the "Bellman's Curse of Dimensionalty". Traditionally, this problem was a concern almost exclusively when optimizing hydrothermal systems through the BR method. Today, with the accelerated addition of many renewable energy sources in power systems, this has become a worldwide issue [2], [3], [4].

One of the seminal works addressing the aforementioned problem is [5]. There, a method known as Stochastic Dual Dynamic Programming (SDDP) was developed and applied to the Brazilian system with great success. This method works well in systems where the operation is mostly deterministic. However, in systems where random events have a strong effect, the SDDP method is not effective as the variance in the estimation of future costs (which are obtained by simulating future scenarios up to a certain horizon) becomes too large. This problem was addressed in [6] to some extent, but the proposed solution is insufificient in our scenario.

Another strategy to solve the SDP problem approximately is known as Rolling Horizons (RH) [7]. This strategy is effective for handling sources in a relatively short time horizon where forecasts (wind, solar) are reliable. However, in systems including reservoirs capable of storing energy for months or years, the aforementioned method is far from optimal.

In the last decade, the application of automatic learning or machine-learning techniques have been applied in almost all areas of engineering and the optimal operation of dynamic systems is no exception [8], [9], [10], [11], [2], [3] entirely devoted to this subject.

A. Contributions

In [12], after comparing different Approximate Dynamic Programming based on machine-learning alternatives, the authors conclude that "*none of these techniques works reliably in a way that would scale to more complex problems*". Our work challenges this view by developing an algorithm capable of operating a complex electrical energy system in a satisfactory way by learning an optimal Operation Policy (OP) through a reinforcement learning loop. Concretely, we address the above issues, mainly the large variance involved in cost estimation, and the complexity in modeling the cost function, by the following techniques, which will be developed later: a) a novel method for choosing the initial states of the simulated trajectories and exploring dynamic; b) using of Common Random Numbers (CRN) as a variance reduction technique; c) modeling the

The present work was possible thanks to the financing received from ANII of Uruguay for the development of the project: FSE_1_2017_1_144926 "Investment planning with variable energies, network restrictions and demand management".

cost function using spatial differences and d) regularizations of parameters over time.

This work is an extension of [13], where the authors presented a success case of learning the optimal operation of a simplified version the energy system of Uruguay. In the present work, we scale the problem to the likely scenario of what the Uruguayan system will look like in 2050 with the the same hydroelectric subsystem, the addition of more solar and wind sources and battery banks, and all petroleum-derived fuel-fired generators removed.

II. Preliminaries

A. Problem setting

The dynamic of the power system is modeled as in (1), where k is an integer that identifies the time-step, X_k is the state-vector of the system at the beginning of the step k, r_k is the vector of non-controlled inputs (rainfall, wind, etc.) and u_k is the vector of controllable inputs (typically the power to be delivered to each generation unit, or power line).

$$X_{k+1}=f(X_k,r_k,u_k,k) \quad (1)$$

The function sc in (2) represents the *cost of operation* during the step k, typically computed as the sum of the fuel consumed by the thermal generators, the imports minus the exports, and any other operation cost including the cost of rationing, in the event that not all the energy demand is fulfilled,

$$sc_k=sc(X_k,r_k,u_k,k) \quad (2)$$

The *operation policy* OP (3) is a mapping that assigns a control vector u_k to different values of the system state and the non-controlled variables at step k,

$$u_k=OP(X_k,r_k,k) \quad (3)$$

Below we define the *state-value or Future Cost function* $FC(X,k)$. This function represents the *expected* cost of the future operation, beginning at state X in the time step k, for a given OP. The *Optimal Operation Policy,* at the step k is the one that minimizes the expected value of the sum of (2) and $FC(X_s,k+1)$ and corresponds to the solution of the optimization problem:

$$FC(X,k)=\left\langle \min_{u} \left[sc(X_k,u_k,r_k,k)+FC(X_s,k+1) \right] \right\rangle_{r_k} \quad (4)$$
$$@ \begin{vmatrix} u \in \Omega(X_k,r_k,k) \\ X_s=f(X_k,u_k,r_k,k) \end{vmatrix}$$

where Ω is the space of possible actions to be taken at step k. Notice that equations (1)-(3) assume that the non-controlled inputs are known at the beginning of each time-step, but nothing is assumed about the future of them.

The non-controlled inputs are modeled as random processes without memory (white noise) with given distributions. If the random processes need to be modeled with memory, a corresponding model with its state variables is incorporated in (1) and fed with corresponding memory-less random variables included in the vector r_k. Notice that whether or not to consider memory in the random processes involved may depend on the time scale of the steps. For example, for a time-step of one hour, the wind power must be represented as a process with memory because the wind does not significantly change at the same time in all wind farms of the country during one hour. Moreover, there is a strong correlation of the wind power between consecutive hours. However, if the time-step is a week or more, representing the wind power as a process with memory does not make sense.

B. The learning loop

Knowing (1) and (2), having an initial estimation of $FC(X,k)$, and a given OP, a number of possible realizations of the operation are simulated; each such realization is called a *trajectory* and depends on the pseudo-random sequence of numbers used to simulate the random inputs. In our case, these simulations are generated using the SimSEE platform (https://simsee.org). Then, a new $FC(X,k)$ is estimated from the information collected by the different simulated trajectories. This recursive estimation is depicted in Fig. 1.

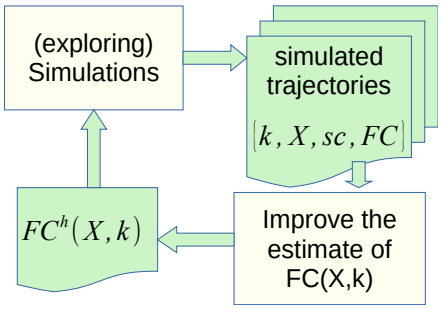

Fig. 1: The learning loop.

More precisely: at the end of the each exploration stage the information collected by trajectories can be represented by: $(X_{ki}, sc_{ki}, FC^h_{ki})$ where $k=1..N_{Steps}$ denotes time step and $i=1...N_{Trajectories}$ is the indicates a particular trajectory. Each trajectory is *determined* by an initial state X_{1i} and a random seed. From the collected information, new values of FC^h_{ki} are estimated follows:

$$FC^h_{ki}=\sum_{p=k}^{p=k+n_{TD}} q^{(p-k)} sc_{ki}+q^{n_{TD}+1} FC^{h-1}(X^s_{(k+n_{TD}+1,i)},k+n_{TD}+1), \quad (5)$$

where q is a *money discount factor,* sc_{ki} is the stage cost defined in (2), and $X^s_{(k+n_{TD}+1,i)}=f(X_{(k+n_{TD},i)},r_{ki},u_{ki},k)$ is next state as defined in (1) and n_{TD} (number of time-difference steps) determines the numbers of sc_{ki} added in the sum of the (5).

III. Modifications to the Forward Method

So far, the learning process described above is known as the standard *forward iteration method*. The main challenge in this method is in estimating FC when the variance of the stochastic elements in the simulation is large, as the number of possible scenarios grows exponentially and deviates wildly, especially if very different future random outcomes are considered. Below we describe the different techniques that we propose for addressing these issues.

A. Initialization of the trajectories

During the exploration stage, simulations are carried out on trajectories starting from a set of initial states of the

system. A set of n_s initial states is obtained by random draws with a distribution that regulates the concentration of the set of states around the true initial state of the system.

At the start of each exploration stage, the sets of initial states and random seeds are randomly chosen and the trajectories, one per each combination of both sets, are initialized.

B. *Representation of the future cost spatial differences*

The information of FC that induces the OP is found in the directional derivatives $\frac{\partial}{\partial X}FC$, note that when determining the control vector in (4) the solution is the same if we add a constant value to FC .Thus, instead of learning FC the proposed algorithm try to learn the differences $FC_{ki}-FC_{kj}$ for each set of new information instead of representing $FC(X_{ki},k)$. In our scenario, the variance of FC_{ki} is orders of magnitude higher than the variance of $FC_{ki}-FC_{kj}$, not having to adjust the model to the FC value allows us to focus the scarce representation resources (model parameters) to represent the relevant information for the OP.

C. *Common Random Numbers (CRN)*

As described before, a key issue is to represent the differences of the state value function when the state changes as a consequence of the control action. Thus, in order for the OP estimation to be reliable, the estimation variance needs to be significantly reduced. The Common Random Numbers (CRN) is a well known variance reduction technique [14] to the estimation of the expected value of the difference of random variable by Montecarlo simulations. We implement the CRN by reusing the same sequence of random inputs, identified by a *seed* for simulating trajectories starting at different initial states described in Section II.A. The different trajectories associated with the same random seed are differentiated by their initial state and the information collected for the groups of trajectories associated with each random seed is treated separately.

D. *State evolution mode*

In order to balance the control of the variance and the exploring capability of the algorithm, the simulation of the trajectories is performed in sections of n_{TD} time-steps following the dynamic of the system (1) chained by steps that we call of *Random Explosion of the State (RES)* where the state of the system is randomly perturbed not following the dynamic restriction (1). The perturbation of the state is carried out with a distribution based on the previous states and their FC estimations that tends to uniformly sample the range of FC values, thus concentrating more samples in the regions of the state space where the directional derivatives $\frac{\partial}{\partial X}FC$ have greater modulus.

Clearly, at each iteration of the learning loop, (5) carries information n_{TD} steps from the future to the present; the longer the time horizon considered, the more iterations will be necessary for $FC^h(X,k)$ to reflect the future consequences of present actions. In this sense, increasing the n_{TD} parameter would seem convenient. In the example case presented, with multi-annual reservoirs, the consequences of the decisions are observed for at least the following 3 years. With an hourly time step simulation, if $n_{TD}=1$, at least 26280 (= 24x365x3) iterations of the learning loop would be necessary for the relevant future information to reach the present at least once. On the other hand, if $n_{TD}=26280$ was set, in one iteration there would already be information on the possible future consequences within 3 years. However, depending on the time constants associated with the state variables, the trajectories associated with the same random seed, although associated with different initial states, converge to a single trajectory, thus losing the ability to collect information on the spatial differences of the state value function. This is why it is important that the n_{TD} value be lower than the smallest time constant associated with the state variables. (e.g.: lower than the emptying time of the lakes for which an operation policy is to be formed).

At the beginning of each iteration of the learning loop, for each of the trajectories associated with a given seed, the first time step where the RES takes place is randomly set in order to blur the effects caused by the partitioning into sections of n_{TD} steps.

IV. Parametric Network Series

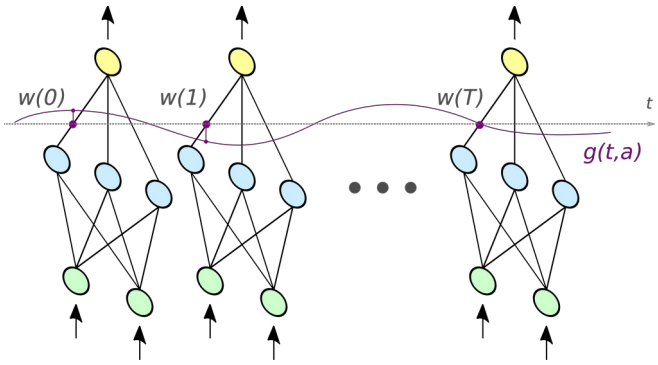

Fig. 2: Parametric network series.

The signals and processes involved in the planning of energy dispatch usually exhibit smooth regular patterns, and so does the cost function. This can be exploited by using parsimonious cost function approximations that can be extrapolated reliably to unseen states. Our proposed method combines the flexibility of Neural Networks (NNs) with prior information about the problem. In a nutshell, the value function, which is a function of state and time, is approximated by a time-step neural network. The architecture of the network is the same for all time steps, as depicted in Fig. 2, reflecting the fact that the structure of the system itself does not change abruptly. The parameters vary across the networks, although in a controlled way: the variation of each parameter is penalized during the training process. The estimate of the value function of iteration h is represented as:

$$FC^h(X,k)=M(X,k,\theta_k)\,, \qquad (6)$$

where θ_k are vectors of fitting parameters for our model that are trained by minimizing the following loss function:

$$L=\sum_{k,g}L_{kg}+\lambda\sum_{k}\|\theta_k\|^p+\beta\sum_{k=2}\|\theta_k-\theta_{k-1}\|^2\,, \qquad (7)$$

where the elements of the first sum have the expression:

$$L_{kg}=\frac{1}{4N^2}\sum_{i\neq j\in g}((M(X_{kj},\theta_k)-M(X_{ki},\theta_k))-(FC_{kj}-FC_{ki}))^2 \quad (8)$$

where g is the set of indexes that identify the trajectories associated with each random seed. As already mentioned, the information collected during the simulation is used to adjust the model based on the spatial differences of the value function associated with the same random seed.

The second summation in (7) adds regularization to the model parameters, with a strength controlled by an hyperparameter λ.

Finally, the third summation limits the abrupt variation of the model parameters with the passage of time; this is controlled by an hyperparameter β.

The learning starts with a Null Policy (NP), $FC^1(X,k)=0$ for all values of state and time. This would be the Operation Policy with zero derivatives in all directions of the state space. The NP is not as naïve as it may seem at first glance: the operating restrictions of the hydroelectric lakes are represented in the dispatch problem as restrictions with penalties for going below certain levels, which put at risk the availability of power from hydroelectric plants, and by restrictions that penalize the operation at high levels due to the effects of flooding of the lake on the surrounding lands. The penalties are established in MUS$/(m.day) which leads to the fact that even with the NP, the operation is generally within the reasonable operation zones.

The β parameter is useful tool for system operators to soften the control actions indicated by the operation policy . As an example, we do not expect the value of water in a reservoir to change radically from one hour to the next. This type of regularization on the parameters can be introduced by the model structure organized in a time-step model.

V. Example Case

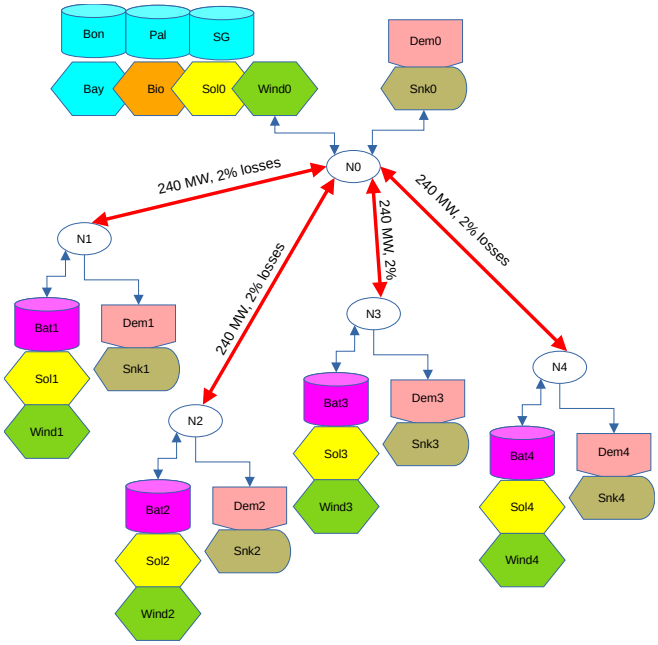

Fig. 3: Diagram of the system of 5 interconnected nodes

As a case study, the Uruguayan electrical system projected to the year 2050 was used. Uruguay is one of the countries with the highest integration of renewable energies [15] and this is expected to increase, as Uruguay does not have oil or natural gas deposits. Concretely, the model was adapted to this scenario by increasing the energy demand to 18.92 GWh, deleting the fuel fired generator, installing 2400 MW and 3200 MW of wind and solar capacity respectively, 400 MW of biomass-based generation, and 32 battery banks of 20 MWx4h each. The Demand and the generation was distributed in five nodes interconnected by 240 MW lines as shown in Fig.3. The 32 battery banks were considered independent operable units, 8 installed in each of the nodes N1-4. Batteries were not installed in node N0. Simulations were generated using the SimSEE platform [13], which is used by the system operator ADME to decide the hourly units commitment of the system.

Table I summarizes the Peak-Demand and installed capacities and their distribution in the system nodes and Table II the corresponding storage capacity.

TABLE I. Installed Cadpacity

		N0	N1	N2	N3	N4	Total
Peak-Demand	[MW]	1492	373	373	373	373	2984
Wind	[MW]	1200	200	200	400	400	2400
Solar	[MW]	160	700	700	100	100	1760
Biomass	[MW]	400					400
Battery - Banks	[MW]		640	640	640	640	2560
Hydro-Baygorria	[MW]	108					108
Hydro-Bonete	[MW]	155.2					155.2

TABLE II. Installed Storage Capacity

		N0	N1	N2	N3	N4	Total
Battery - Banks	[GWh]		2.56	2.56	2.56	2.56	10.24
Hydro-Baygorria	[GWh]	6.18					6.18
Hydro-Bonete	[GWh]	1101.56					1101.56
Hydro-Palmar	[GWh]	84.67					84.67
Hydro-Salto Grande	[GWh]	62.96					62.96

The dimension of the state space of the system is 9; three dams for water storage, four battery banks, and two river runoffs. To fix ideas, if each dimension was discretized into 10 values and the Bellman Recursion was used to resolve the above system over 1.5 years, $10^9 \times 365 \times 1.5 \times 24 = 1.31 \times 10^{13}$ energy dispatch problems would need to be solved.

The chosen neural network architecture is depicted in Fig.2 and consists of a hidden layer of 16 neurons followed by an output layer of one neuron, both with hyperbolic tangent as saturation function. Lasso regularization [16] ($p=1$ in (7)) was used on the parameters with weight $\lambda=1E\text{-}8$.

VI. RESULTS AND DISCUSSION

Fig. 4 shows the future cost of the operation evaluated on the same 100 realizations of the stochastic processes starting from a known state of the system. Each value in the horizontal axis corresponds to an iteration in the algorithm, and the corresponding vertical coordinate shows the total cost incurred in operating the system using the policy obtained at that iteration. The value x=1 corresponds to the initial null policy. As can be seen, the performance decreases after the first iteration, but consistently improves afterwards. This behavior is to be expected and is consistent with what was mentioned in the previous section, as propagating information to the present takes some time.

The 4000 iterations of the learning loop shown in Fig.4 took 73 hours on a 48 threads computer. This computational cost is necessary only for initial learning. In the continuous-time application on the real system, every hour elapsed the

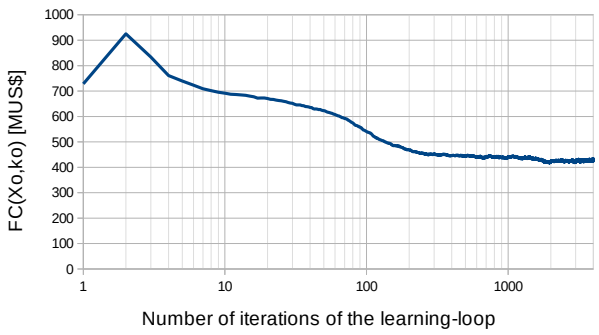

Fig. 4: Convergence of the evaluation of the Policies learned

neural networks shown in Fig. 2 are shifted to the left, eliminating the first one and duplicating the last one, and 1 hour (54 iterations) is available to improve the FC representation. If it is necessary to carry out more iterations, the algorithm is easily parallelizable since the simulations of the exploration stage are totally independent processes.

VII. CONCLUSIONS

A successful case was presented in the determination of an Operation Policy of a generation system with 9 state variables and very different time constants. In this system, the application of the Bellman Recursion is not applicable. Given the need to adequately represent the correlations between stochastic processes, which involve very different time constants, strategies such as SDDP or Rolling Horizons techniques are not directly applicable either.

The success achieved is based substantially on two keys: a) The use of CRN to control the variance of the differences of the value function and b) On the controlled exploration based on trajectories formed by sequences of steps in which it allows the system to follow its natural evolution interrupted by random jumps in the state that rebuild the exploration capacity.

One line of research to improve the learning speed could be to measure the variance in each of the state directions as the system evolves, as a way of estimating the exploration capacity of the set of trajectories associated with each random seed. With this measure, when the reduction of the exploratory capacity is greater than a given threshold, the random explosion of the states would be applied to that set of trajectories, thus recovering the exploration capacity.

VIII. DISCLAIMER

The content of this article is entirely the responsibility of its authors, and does not necessarily reflect the position of the institutions of which they are part of.

REFERENCES

[1] R. Bellman, *Dynamic programing*. Princeton University Press, 1957.

[2] W. B. Powell, *Approximate dynamic programming*. Wiley, 2011.

[3] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[4] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, Second edition. Cambridge, Massachusetts: The MIT Press, 2018.

[5] M. V. F. Pereira and L. M. V. G. Pinto, "Multi-stage stochastic optimization applied to energy planning," *Math. Program.*, vol. 52, no. 1–3, pp. 359–375, May 1991, doi: 10.1007/BF01582895.

[6] M. P. Soares, A. Street, and D. M. Valladão, "On the solution variability reduction of stochastic dual dynamic programming applied to energy planning," *Eur. J. Oper. Res.*, vol. 258, no. 2, pp. 743–760, Apr. 2017, doi: 10.1016/j.ejor.2016.08.068.

[7] É. Cuisinier, P. Lemaire, B. Penz, A. Ruby, and C. Bourasseau, "New rolling horizon optimization approaches to balance short-term and long-term decisions: An application to energy planning," *Energy*, vol. 245, p. 122773, 2022, doi: https://doi.org/10.1016/j.energy.2021.122773.

[8] Z. Zhou, L. Zhang, F. Zhu, W. Tu, Z. Yuan, and X. Gou, "Dynamic compensation pricing scheme for demand resources based on deep reinforcement learning," in *2020 IEEE/IAS Industrial and Commercial Power System Asia (I&CPS Asia)*, Weihai, China, Jul. 2020, pp. 138–143. doi: 10.1109/ICPSAsia48933.2020.9208435.

[9] K. De Vos, N. Stevens, O. Devolder, A. Papavasiliou, B. Hebb, and J. Matthys-Donnadieu, "Dynamic dimensioning approach for operating reserves: Proof of concept in Belgium," *Energy Policy*, vol. 124, pp. 272–285, Jan. 2019, doi: 10.1016/j.enpol.2018.09.031.

[10] A. Nayak and L. Heistrene, "Hybrid machine learning model for forecasting solar power generation," in *2020 International Conference on Smart Grids and Energy Systems (SGES)*, Perth, Australia, Nov. 2020, pp. 910–915. doi: 10.1109/SGES51519.2020.00167.

[11] T. Zhao, J. Wang, X. Lu, and Y. Du, "Neural Lyapunov control for power system transient stability: A deep learning-based approach," *IEEE Trans. Power Syst.*, vol. 37, no. 2, pp. 955–966, Mar. 2022, doi: 10.1109/TPWRS.2021.3102857.

[12] D. R. Jiang, T. V. Pham, W. B. Powell, D. F. Salas, and W. R. Scott, "A comparison of approximate dynamic programming techniques on benchmark energy storage problems: Does anything work?," in *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, Orlando, FL, USA, Dec. 2014, pp. 1–8. doi: 10.1109/ADPRL.2014.7010626.

[13] R. Chaer *et al.*, "Teaching a robot the optimal operation of an electrical energy system with high integration of renewable energies," in *2021 IEEE URUCON*, Montevideo, Uruguay, Nov. 2021, pp. 364–367. doi: 10.1109/URUCON53396.2021.9647311.

[14] J.-J. Christophe, J. Decock, J. Liu, and O. Teytaud, "Variance reduction in population-based optimization: application to unit commitment," in *Artificial Evolution*, vol. 9554, S. Bonnevay, P. Legrand, N. Monmarché, E. Lutton, and M. Schoenauer, Eds. Cham: Springer International Publishing, 2016, pp. 219–233. doi: 10.1007/978-3-319-31471-6_17.

[15] E. Cornalino *et al.*, "Handling the intermittence of wind and solar energy resources, from planning to operation. Uruguay's success," in *36th USAEE/IAEE North American Conference, Washington, DC*, Sep. 2018, pp. 23–26.

[16] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*, 2nd ed. Springer, 2009. [Online]. Available: http://www-stat.stanford.edu/~tibs/ElemStatLearn/