CS4210-B: Electric Vehicles Charging Planning with Reinforcement Learning

Douwe den Blanken 4843940
Frans de Boer 5661439
Adarsh Denga 4780175
Sjoerd Groot 4694368
Laurens Krudde 4714288
Grigorii Veviurko Supervisor

September 19, 2023

Abstract

The Electric Vehicle Charging Planning (EVCP) is an important problem in the context of sustainable energy. In this paper, we attempt to combine the two components of reinforcement learning and constraint-based solving to arrive at a solution for the EVCP problem. We use a simulation environment as well as our own implementation of the TD3 algorithm along with constrained projection. Through our experiments in learning over single and multiple days with and without considering constraint violations, we learn that while learning non-linear power constraints is challenging, we are able to maximise the utility of the network if we do not take into account constraint violations.

1 Introduction

The European Union has made a proposal prohibiting the sale of Internal Combustion Engines (ICE) vehicles in the year 2035 [1]. As a result, the number of electric cars will increase significantly. While this is a great step towards reducing our influence on the climate, it remains to be seen if our electricity grid in its current state is able to supply electricity to all these electric cars.

Renewable energy sources (RES), like solar energy and wind energy, are also finding their way into today's power grids at rapid speed. Renewables are set to account for almost 95% of the increase in global power capacity through 2026, with solar power alone providing more than half [2]. While also being highly beneficial in reducing our current climate change rate, these RES do add more complexity to power grids as their power supply is not fully reliable. Together with the varying demand at charging stations, distributing power in a grid with limited supply becomes an interesting and important puzzle to solve.

On the plus side, electric vehicles can be flexible about when their demand is served and therefore allows for the opportunity to plan their charging. This research aims to find a solution to the problem of Electric Vehicle Charging Planning, from here on called EVCP, using deep reinforcement learning.

The EVCP problem is hard to tackle, as it is non-convex [3] and highly stochastic. The stochastic component originates from the fact that EVs arrive at charging stations seemingly randomly. Furthermore, the solution to the problem has to adhere to physical constraints.

This study aims to develop a Reinforcement Learning (RL) based method to approach the problem. It is evaluated against two simple baselines - a greedy heuristic and a deterministic planner with perfect knowledge of the future. While both of these more classical approaches have no problem adhering to the constraints, due to the fact that their solvers can take these directly into account, they are unable to handle the stochasticity. The RL agent on the other hand can better deal with this, but there the challenge lies in adhering to said constraints.

In this research, the IEEE 16 bus system [4] is used as an environment to train a Twin Delayed Deep Deterministic Policy Gradient [5] (TD3) agent. It is shown that the agent is able to learn to charge the EVs in the grid and outperform the greedy algorithm when disregarding power grid constraints. Ways to account for these power grid constraints have been explored by penalizing constraint violations and by projecting the model output into the feasible region. Lastly, as learning the constraints proved to be difficult, the agent was restricted to only predicting the power with which to charge the EVs and leave solving the constraint problem to the greedy heuristic with added constraints based on the prediction.

The paper is structured as follows. Section 2 gives the necessary background information. Section 3 examines related work and existing solutions. Section 4 explains the used deep reinforcement learning model. Section 5 explains the used dataset. Section 6 gives the executed experiments and their results. Finally, section 7 gives a conclusion and section 8 proposes ideas for future areas of research.

2 Background

In this section, some of the relevant background information for this paper will be highlighted.

Optimal Power Flow (OPF)[6]: The optimal power flow problem is a very relevant problem in power systems. The objective of the OPF problem is to find a steady-state operating point of a network that minimises the cost of electric power generation while also adhering to a set of constraints that relate to the operational limits of the network as well as the demands of devices in the grid. The EVCP problem in this paper can be seen as a multi-timestep OPF problem, where the coupling between timesteps is embodied in the State of Charge (SoC) dynamics of the EVs.

Electric Vehicle Charging Planning (EVCP) [6]: In the EVCP problem, the standard power grid from the classical OPF is extended by the addition of electric vehicle charging stations. This adds stochasticity to the grid as demand varies due to arrival and departures of the EVs. However, unlike conventional loads, the electric vehicles (EVs) can be flexible about when their demand is met, which gives rise to the planning problem. Furthermore, it is assumed that the vehicles cannot always be charged fully and therefore each EV has a linear utility function to quantify how much value is added by charging the EV. This research considers the setting from [6] where the combined interest of all EVs is considered. The goal is therefore to maximize the sum of demand met factored by the utility coefficient over all EVs. This is also referred to as social welfare. Social welfare is to be maximized while satisfying the power flow constraints.

Next to the loads, both conventional and EV-based, there are also generators present in the grid. Regular generators have a cost defined for the power they supply, while renewable energy sources (RES) do not have costs associated with them. Both the costs of the power supplied by the generators and the amount of power supplied by the RES are stochastic.

Formally, the objective of the EVCP, which has to be solved at each timestep, can be stated as:

$$\max_{\mathbf{v},\mathbf{p}} \quad J(\mathbf{p}) = \sum_{\mathbf{t}} \langle \mathbf{p^t}, \mathbf{u^t} \rangle \tag{1}$$

where $\mathbf{p}, \mathbf{v}$ are the nodal power and voltages per time step. $\mathbf{u}$ are the utility coefficients; this includes the importance of charging for each EV, but also the cost coefficients of the generators. Power at the generators is negative, as the grid is taking power from it. Power at the loads is positive, as the grid is supplying power to it. Thus meeting supplied demand should be maximized while costs should be minimized.

Furthermore, the problem is subject to certain constraints. First, the constraints related to the operational limits of the grid can be given by:

$$\underline{\mathbf{P^t}} \leq \mathbf{p^t} \leq \overline{\mathbf{P^t}} \tag{2}$$

$$\underline{\mathbf{V^t}} \leq \mathbf{v^t} \leq \overline{\mathbf{V^t}} \tag{3}$$

$$p_i^t = -v_i^t \sum y_{ij}(v_i^t - v_j^t) \tag{4}$$

$$-\bar{\mathbf{I}}_{\mathbf{ij}} \leq \mathbf{Y_{ij}}(v_i^t - v_j^t) \leq \bar{\mathbf{I}}_{\mathbf{ij}} \tag{5}$$

The power $\mathbf{p^t}$ and voltage $\mathbf{v^t}$ assigned to a node must fall within the limits of a node, as specified by equations (2) and (3). Furthermore, the power assigned to a node must match up with the difference in power flowing into a node and flowing out of a node (4), where y_{ij} denotes whether nodes i and j are connected. Lastly, the line currents as a result of the assigned node voltages and network impedance may not exceed the maximum or minimum line current (5).

Secondly, the EVs are subject to the following constraints:

$$e_k^{t_k^{arr}} = 0 \tag{6}$$

$$0 \leq e_k^t \leq \overline{E}_k \tag{7}$$

$$e_k^{t+1} = e_k^t + \Delta t p_k^t \tag{8}$$

where e_k^t is the state-of-charge (SOC) of EV k at time step t and $\overline{E}_k$ is the desired SOC. The SOC is taken to be 0 upon arriving at a charging station, as formulated in constraint (6). Constraint (7) ensures that the SOC of an EV should always be at least 0 and cannot exceed the desired SOC. Furthermore, constraint (8) states that the SOC of an EV in the next time step is increased exactly by the amount it was charged in the previous time step.

The problem is essentially an optimization problem. Unfortunately, it is, like OPF, nonconvex. This means there is no guarantee of obtaining the global optimum. Furthermore, the problem may become intractable for large grids. Also, the problem changes on a fast timescale due to the stochastic components like the power supply from RES and the arrival and departure times of the EVs at the charging stations. This requires a lot of resolving and may be intractable for longer planning horizons.

Markov Decision Process: In a Markov Decision Process (MDP) the next state depends only on some sufficient metric of the previous states. In the environment used in this research, however, there exists inherent stochasticity due to three factors. Firstly, the power output of the RES depends heavily on the weather, i.e. the amount of sun the solar panels receive. Secondly, the price of power supply from the generators changes over time. Thirdly, the arrival and departure times of the EVs themselves introduce stochasticity. As such, the problem is modelled as an MDP, where the current state is fully known, but the future cannot be completely determined by the past.

Constraints: The main problem with solving the charging planning problem with reinforcement learning is that there is no trivial way of incorporating constraints into a reinforcement learner. Individual actions, being the power and voltages supplied to each node, can be bound between some values by the network. However, the problem are the more complex constraints, being functions of the nodal power and voltages, that are to be satisfied in order to comply with the laws of physics.

Solvers: Three existing solutions were used as simple baselines to compare to. First, a solver using greedy heuristics, by providing as much power as possible without violating constraints, is used to yield a 'greedy' solution. Secondly, providing all data of the episode to the solver, such that it knows the future, yields a 'deterministic' solution. The latter indicates the maximum obtainable reward while satisfying all constraints (no violations). Finally, a solver was made that always attempts to give the maximum amount of power to all devices without taking constraints into account. This solver achieves the highest possible reward for an episode. In general, the goal is to outperform the greedy solver such that the agent's solution is between the greedy and deterministic solution.

3 Related work

In [6], the concept of using DC in distribution grids to reduce power losses and the amount of required equipment is introduced. Furthermore, the challenges of connecting EVs to this grid are introduced and the mathematical groundwork for this optimization problem is laid out.

Several other papers have proposed solutions to solve either the OPF problem or the EVCP problem. More specifically [7] and [8] both introduce solutions to the OPF problem by using end-to-end learning and deep neural networks. A possible issue with modelling this problem as an end-to-end learning problem is that it does not take the agent's interaction with the environment into account, which is why this research opts to use RL to solve the EVCP problem.

The papers [9], [10], and [11] attempt to solve the EVCP Problem using Reinforcement Learning. However in [9] the constraints of the electricity network are not followed, and they do not include stochastic variables such as energy production by solar panels. The network used in [9] is a multilayer perceptron (MLP), and is thus unable to use past information to make predictions. Zhen et al. [10] apply TD3 to solve the EVCP Problem. To teach the RL model to adhere to the constraints their reward function penalizes if not all the constraints are met and only rewards based on how optimal the power flow is if they are met. They do not use any RES in their research. Li et al. [11] apply DDPG to find optimal power flow while also maximizing the use of RES. They model the problem as a multi-objective optimization problem, where they try to minimize the power generation costs, the voltage fluctuation, and the power transmission loss. All three of these researches attempt to solve the same problem as this paper in different ways and can be used as a basis to work from.

Resource-Constrained Deep Reinforcement Learning [12] uses DDPG to create a system to allocate limited resources. The algorithm is used to allocate ambulances to base stations and to allocate shared bikes to docking stations. The concept of resource-constrained deep RL can be applied to the EVCP problem, where the number of cars that could be charged at any given moment might be limited.

4 Model setup

The underlying model that is used throughout the experiments is based on a custom TD3 implementation built in PyTorch. Although initially a pre-made implementation of TD3 was used [1], it was finally opted to go for a custom implementation due to lack of flexibility in the framework.

The architecture of the actor used in the model can be varied, but it generally supports any number of LSTMs followed by any number of linear layers, either of which can also be zero. The general architecture is illustrated in figure 1. The number of layers and the size of the hidden dimensions of the LSTMs and linear layers are taken as hyperparameters in the experiments. The number of layers considered are in $\{0, 1, 2, 3, 4\}$ and the hidden dimensions in $\{32, 64, 128, 256\}$.

The model has as input an observation of the grid at the current time step. This is a vector that contains for each device the lower power bound, the upper power bound, the lower voltage bound, the upper voltage bound and the utility value. Such that the input to the model is a vector of length 5 times the number of devices. The power grid used in this research contained 22 devices such that the size of the input vector is 110.

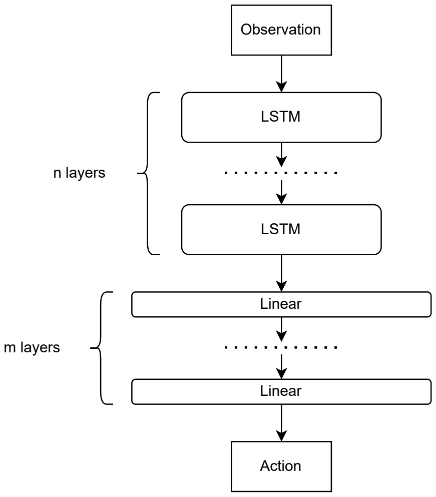


Figure 1: Network architecture for actor in the TD3 model.

Based on the observation, the model will output the power and the voltage for each device in the grid. So the action is a vector with a length of twice the number of devices, in this case, 44. The action values from the model are in the range $[-1, 1]$. These values are rescaled to real power and voltage values by mapping $[-1, 1]$ to the range between the lower and upper bound for the power and voltages for each node. In this way, constraints (2) and (3) from section 2 are trivially satisfied.

The environment also has toggleable observation normalization which scales all observations to be between 0 and 1.

5 Data origin

As mentioned in section 2, the power grid contains some stochastic components, namely the power supply price, the renewable energy source power supply and the arrival and departure times of the EVs at the charging stations. In order to simulate this behaviour, real-world data was used as a basis.

However, there is not one dataset which encompasses all of the information that we need to simulate this problem. Therefore, multiple datasets are combined to enable the environment to sample all of the required data.

[1] https://docs.ray.io/en/latest/rllib/rllib-algorithms.html#ddpg

For example, data from the Pecan Street[2], yielded, per house, the grid power used, solar power generated, electric car power usage and total power usage (per minute, from multiple days). Data from the National Institute of Standards and Technology[3] was used for AC and DC-generated powers. For energy prices, a dataset containing electricity prices from 1st of January 2019 to 31st of March 2019 was used. Finally, data from ElaadNL[4] was used for the arrival time, departure time, duration and energy charged per electric car.

Due to the fact that this last data set does not contain a lot of samples, the average arrival rate curve was extracted from this data and used in a Poisson process to simulate the arrival times: the duration and demand were sampled from the same data set for sessions with relatively similar arrival times.

6 Experiments

This section will give an overview of the experiments that were run and the obtained results.

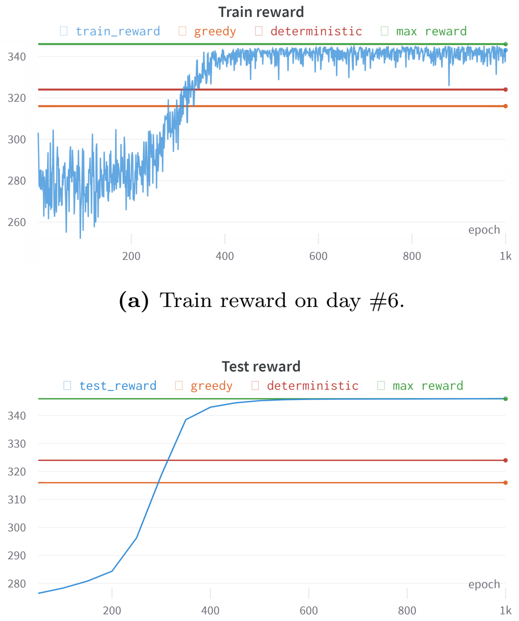
6.1 Benchmarks

As mentioned in section 2, the EVCP problem is an optimization problem and can also be solved using (non-linear) solvers. Such solvers were used to obtain benchmark values. The three solvers mentioned in section 2 were used to compare against.

6.2 Learning a single day without constraints

Starting off, to confirm whether the entire system was working, initial tests were performed on a single deterministic episode from the environment to induce overfitting (memorisation). This way, if the reward would increase over epochs, it could be verified that the model is actually learning.

For this experiment, the environment was configured to not give penalties for violating constraints so the learner is able to achieve the maximum reward. The experiment used an actor network architecture of two LSTM layers with hidden size 128 followed by four linear layers of size 128. Furthermore, every epoch in training was performed on the same episode #6 with the same seed (for reference).



(a) Train reward on day #6.

(b) Test reward on day #6.

Figure 2: Demonstration of overfitting the agent on a single episode. The green, red and orange lines show the max, deterministic and greedy rewards respectively.

The result of this experiment can be found in 2. It can be seen that the agent is able to fully 'learn' the environment by clamping to the max reward. In figure 2a, it can be seen that the training reward shows stability near the maximum reward level, but does not exactly reach the max due to exploration noise, which is also the cause for the graph to look 'noisy'. The test reward, shown in figure 2b, does reach the maximum reward after epoch 600. Note that testing was done on the same single day as used for training.

It remains to see how the overfitted agent performs on other days. Figure 3 shows the test reward of the agent on different days. It must be noted that each day has a different possible maximum reward, and also the greedy and deterministic rewards will be different. The rewards are therefore normalized between the greedy and maximum reward to make the plot readable. It can be seen that the agent is no longer able to reach the maximum reward and even under-

[2] https://www.pecanstreet.org/

[3] https://pvdata.nist.gov/

[4] https://elaad.nl/en/

performs the greedy heuristic, which gets worse the more it is overfitting.

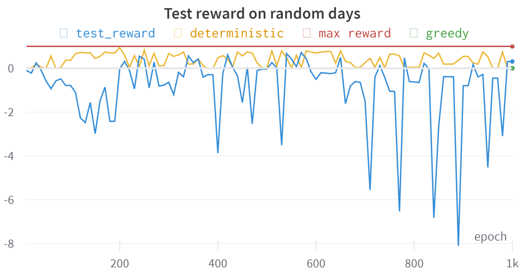


Figure 3: Performance of the overfitted agent on random days. Rewards are normalized between the maximum reward and the greedy reward.

6.3 Learning random days without constraints

The next experiment was intended to validate that it could learn without constraints on the entire dataset instead of only overfitting on a single day. Violating current or power constraints were still left out of the reward function.

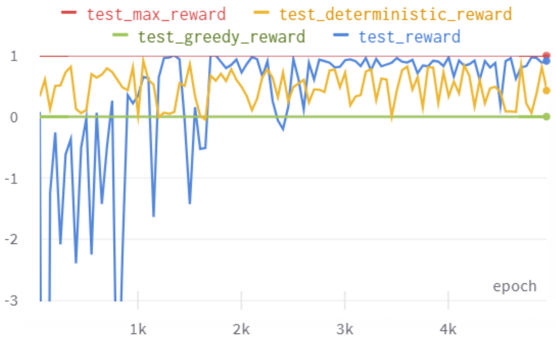


Figure 4: Performance when training on random days, normalized between the maximum reward and the greedy reward.

In Figure 4 the test reward obtained by the model is again normalized with the greedy reward and the max reward. It can be seen that the agent is outperforming both the greedy and the deterministic solvers. The latter of which is only possible because the agent is ignoring the power grid constraints. Table 1 illustrates the severity of the power flow constraint (3) and current flow constraint (4) violations made by the agent.

	Average power flow constraint violation	Average current flow constraint violation
Greedy solver	300	0
TD3 agent	10000	15000

Table 1: Average constraint violation by way of solving EVCP.

6.4 Penalizing power constraint violations

In the previous two subsections, it is shown that the agent is able to learn to charge the EVs. The next step is to reduce and preferably omit the constraint violations such that it actually produces feasible solutions. The first intuitive approach is to penalize the agent for violating the power grid constraints. This way, the agent should learn to produce actions that are within the feasible region. To test if the agent is able to learn to avoid constraint violations at all, it was run in a single day just like the first experiment.

For the training reward, the model was given the negative of the sum of power violations for all grid nodes. Here, a power violation is measured as the difference in power between the power predicted that the generators and EVs need to produce or consume and the actual power as a result of the predicted node voltages. The utility and current constraints were left out during this experiment to isolate the learning of the power constraints. As the actor and critic model, an MLP was used with 4 layers and a dimension of 256.

As shown in Figure 5, when training the agent, the number of violations did decrease. However, it never got below the total amount of power that should be flowing in the system let alone go to 0. In addition, it was observed that there was a big discrepancy between test and train violations. The discrepancy was most likely the result of the policy noise, as policy noise is only added during training to encourage exploration and not during testing.

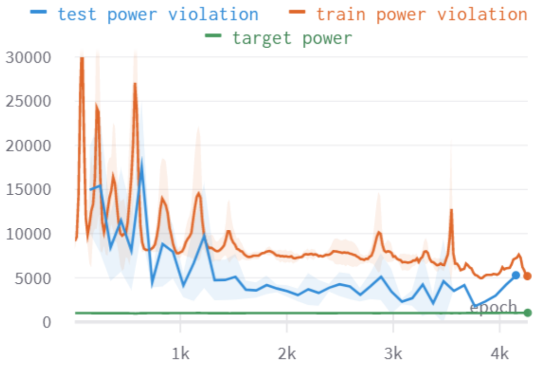


Figure 5: Average power flow violations of 5 runs with a policy noise of 0.05.

6.5 Constrained projection

Constrained projection attempts to minimize the (euclidean) distance between an input solution (that might violate constraints), and a new solution that violates no constraints. This can be written as:

$$\min \sqrt{\sum_{i=0}^{n-1} (x_i - y_i)^2} \tag{9}$$

Where x is the input solution, and y is the output solution. In the case of power and voltages x would be:

$$x = [p_0, p_1, ..., p_{n-1}, v_0, v_1, ..., v_{n-1}] \tag{10}$$

Where n is the number of devices. A linear solver is used to find a solution to this equation.

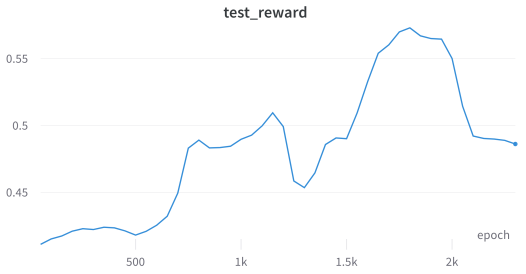


Figure 6: Training reward achieved using constrained projection

The results of this experiment can be seen in figure 6 and 7. Figure 6 shows the reward, which at all times is very close to 0. Figure 7 shows the power violations at each time step, which is always near 0.

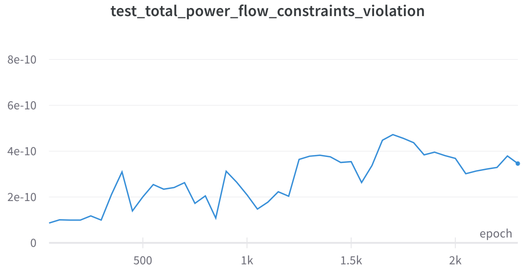


Figure 7: Power violations achieved using constrained projections

It can be seen from these plots that constrained projection manages to minimize the number of constraint violations, but it does this at a heavy cost to the reward. One possible explanation of this is that the model never learns to output power and voltages that the solver can work with, which causes the best solution the solver can find to be a near-zero power solution. Because the solver does not take reward into account (only the distance between 2 solutions) it finds this near-zero reward optimal and returns that.

6.6 Predicting EV max power

As learning non-linear power and current constraints proved to be difficult the next approach tried was only predicting the maximum power with which to charge the EVs. The model outputs a new upper bound for the EV charging power used by a greedy solver to maximize the utility. The hypothesis would be that the model would learn when not to charge the EVs to make use of renewable energy at a later point in time. To validate that this was feasible to train the model was again trained on a single day. From Figure 8 it becomes clear that although the model improved upon its initialization, it did not achieve to match, let alone exceed the greedy solver.

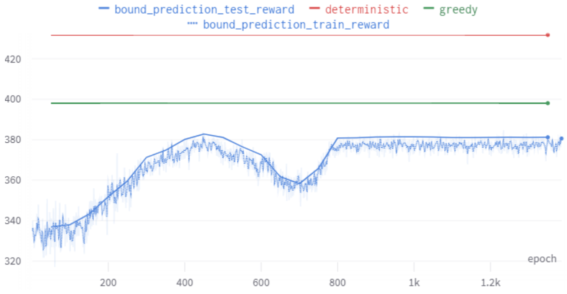


Figure 8: Utility of the model predicting EV charging upper bound

7 Conclusions

In this paper, we investigated the use of Reinforcement Learning to solve the Electric Vehicle Charging Planning Problem. From this, we can conclude the following:

- When ignoring power grid constraint, a TD3 agent is able to learn to charge EVs in the power grid.
- Without gradients, it is very hard to learn the non-convex relation between power and voltage
- Constrained projection, where a linear solver tries to minimize the distance between a given solution and a valid solution, generates solutions that have no constraint violations, but this comes at the cost of a severely reduced reward if the model output has not properly learned what powers would be feasible.
- A hybrid approach, where RL predicts how much power the EVs need, leaving the power grid solving to a secondary system seems to work better as it is guaranteed to provide a feasible solution

We hope that this body of work guides future research into the EVCP problem by documenting which RL-based approaches show promise and which other methods are probably a dead end.

8 Future Work

Due to time constraints, this research was not able to investigate every possible avenue of solving the EVCP Problem. Several things could be researched in future work:

8.1 Providing time as an input

The model in this research does not have an input for what the time of the day is, or what day in the year it is. Theoretically, the agent could derive this from the power supply from renewable energy sources. However, the idea would be the other way around such that the agent can more quickly learn to predict the power supply from for example solar panels by learning the relation with the time of the day, and even the day of the year to get even more precise.

8.2 Differentiable constrained projection

The ability to learn just about anything using deep learning arguably comes from the backpropagation algorithm. This algorithm allows for a gradient to propagate through the layers of a network, making adjustments to the parameters in each layer to make a small step in the right direction of the optimal solution. Constrained projection is non-differentiable, thus backpropagation cannot be used on the output of the constrained projection algorithm to modify the neural network parameters. By applying a differentiable variant of constrained projection as an extra layer on top of a neural network, the neural network and constrained projection algorithm can learn to work together to more efficiently get to good solutions.

8.3 Data from the same origin

The EVCP contains multiple stochastic components; the power supply price, the renewable energy sources and the arrival and departure times of the EVs at the charging stations. These components are simulated based on data sets with different origins, some even from different continents. For the sake of exploratory research, it is not that big of a problem. However, in future research, it would add to the credibility if all the data would originate from one and the same region.

References

[1] Nick Carey and Christoph Steitz. Eu proposes effective ban for new fossil-fuel cars from 2035. https://www.iea.org/news/renewable-electricity-growth-is-accelerating-faster-than-ever-

worldwide-supporting-the-emergence-of-the-new-global-energy-economy, July 2021.

[2] IEA. Renewable electricity growth is accelerating faster than ever worldwide, supporting the emergence of the new global energy economy. https://www.iea.org/news/renewable-electricity-growth-is-accelerating-faster-than-ever-worldwide-supporting-the-emergence-of-the-new-global-energy-economy, December 2021.

[3] Javad Lavaei and Steven H. Low. Zero duality gap in optimal power flow problem. *IEEE Transactions on Power Systems*, 27(1):92–107, 2012.

[4] Jia Li, Feng Liu, Zhaojian Wang, Steven H. Low, and Shengwei Mei. Optimal power flow in stand-alone dc microgrids. *IEEE Transactions on Power Systems*, 33(5):5496–5506, 2018.

[5] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. *Proceedings of Machine Learning Research*, 80:1587–1596, 10 2018.

[6] Grigorii Veviurko, Wendelin Böhmer, Laurens Mackay, and Mathijs de Weerdt. Surrogate dc microgrid models for optimization of charging electric vehicles under partial observability. *Energies*, 15(4), 2022.

[7] Deepjyoti Deka and Sidhant Misra. Learning for DC-OPF: classifying active sets using neural nets. *CoRR*, abs/1902.05607, 2019.

[8] Xiang Pan, Tianyu Zhao, and Minghua Chen. Deepopf: Deep neural network for DC optimal power flow. *CoRR*, abs/1905.04479, 2019.

[9] Nasrin Sadeghianpourhamami, Johannes Deleu, and Chris Develder. Definition and evaluation of model-free coordination of electrical vehicle charging with reinforcement learning. *CoRR*, abs/1809.10679, 2018.

[10] Hongyue Zhen, Hefeng Zhai, Weizhe Ma, Ligang Zhao, Yixuan Weng, Yuan Xu, Jun Shi, and Xiaofeng He. Design and tests of reinforcement-learning-based optimal power flow solution generator. *Energy Reports*, 8:43–50, 2022. 2021 The 8th International Conference on Power and Energy Systems Engineering.

[11] Jinhao Li, Ruichang Zhang, Hao Wang, Zhi Liu, Hongyang Lai, and Yanru Zhang. Deep reinforcement learning for optimal power flow with renewables using spatial-temporal graph information. *arXiv preprint arXiv:2112.11461*, 2021.

[12] Abhinav Bhatia, Pradeep Varakantham, and Akshat Kumar. Resource constrained deep reinforcement learning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pages 610–620, 2019.