Reinforcement Learning for Demand Response of Domestic Household Appliances

Reymond, Mathieu; Patyn, Christophe; Radulescu, Roxana; Nowe, Ann; Deconinck, Geert

Published in:
Proceedings of the Adaptive Learning Agents Workshop 2018 (ALA-18)

Publication date:
2018

Document Version:
Final published version

Reinforcement Learning for Demand Response of Domestic Household Appliances

Mathieu Reymond*
Vrije Universiteit Brussel
Brussels, Belgium
mreymond@ai.vub.ac.be

Christophe Patyn*†
Vrije Universiteit Brussel
Brussels, Belgium
christophe.patyn@vub.be

Roxana Rădulescu
Vrije Universiteit Brussel
Brussels, Belgium
roxana@ai.vub.ac.be

Geert Deconinck
Katholieke Universiteit Leuven
Leuven, Belgium

Ann Nowé
Vrije Universiteit Brussel
Brussels, Belgium

ABSTRACT

With today's electricity grid being penetrated with more and more intermittent energy sources, the need arises to match demand for electricity with electricity generation, in order to maintain the stability of the grid. Demand response has been proposed as a mechanism that aims to solve this problem, by stimulating the shift of electricity demand towards production peaks. In previous work individual devices were trained using fitted-Q iteration to shift the consumption of electrical energy towards low-price periods, while still guaranteeing user comfort. This paper shows that controlling multiple devices with one independent agent per device is more cost-effective compared to a centralized agent. The paper expands the setting with an energy consumption constraint on the level of the household. The objective is to spread the energy consumption of the devices during low price periods as much as possible in order to avoid overloading the grid. Preliminary results for both centralized and independent learners show that these agents are capable of reducing the amount of constraint violations minimally but fail to reduce violations to zero.

KEYWORDS

multi-agent systems; reinforcement learning; demand response

1 INTRODUCTION

Every day, the electric grid is required to deal with more and more distributed sources of variable electric energy generation [18]. This causes electric grids to face a more difficult challenge for maintaining grid stability (i.e., at every point in time, generation of electricity needs to equal consumption). The intermittent character of these sources forces Transmission and Distribution System Operators (TSO and DSO) to shift towards a system that focusses on managing the demand, rather than the supply of electricity, in order to maintain a strict balance between the two. Demand response technologies can be applied to incentivize end users to switch on electric devices when a lot of renewable electric power is generated, and switch them off when generation is low. One example is price-based demand response [1] with a day-ahead market price for electricity. In this scheme, a consumer receives an electricity price profile. This price profile contains a different electricity price for every hour of the next day. It is then the consumer's role to optimize its energy consumption which should result in the shifting of electric loads to low-price periods. When a cluster of consumers buys into these demand response schemes, a large part of the demand for electric energy can be shifted to time periods when high amounts of renewable energy generation are available. In this way, demand response can enable a grid with higher penetration of renewable energy.

With the ongoing electrification of transport and heating in mind, for a single household demand response could result in a heat pump, electric vehicle, electric water heater, cooking stove, etc. all drawing electric power at more or less the same time. Therefore, additional restrictions should be put in place in order to prevent the feeder line of the residence from operating during overcapacity. Intelligent agents can control these devices and cooperate to minimize the global electricity cost of the house while avoiding damage or overheating of the feeder line.

In this paper, the simulated residence will have a combination of interruptible and uninterruptible loads. The interruptible loads are a heat pump (HP) and an electric water heater (EWH). The uninterruptible load is a dishwasher (DW). The operation of each individual device adheres to device-specific constraints. In the second part of this paper, a capacity constraint on the feeder is taken into consideration for the residence as a whole. Two control designs are compared. One solution trains a single reinforcement learning agent that controls all devices at once. The other trains one reinforcement learning agent per device. In the latter case, the technique of difference rewards [30] is applied to coordinate the independent learners indirectly. Both designs use the fitted Q-iteration algorithm (FQI) [12, 21] to train the agents.

The contributions of this paper are the following: (1) it shows that FQI can be applied in a multi-agent setting when combined with difference rewards; (2) it shows that independent agents outperform the centralized agent and learn more cost-effective policies.

2 BACKGROUND

2.1 Reinforcement Learning

Reinforcement learning (RL) [27] is a machine learning technique that allows an agent to learn by trial-and-error while interacting with an environment, given some numerical feedback known as a reward signal. The environment is modelled as a Markov decision

*These authors contributed equally to this work.
†Also with Katholieke Universiteit Leuven.

process (MDP) $M = (S, A, T, \gamma, R)$ [20], where S, A are the state and action spaces, $T\colon S \times A \times S \to [0, 1]$ is a probabilistic transition function, γ is a discount factor determining the importance of future rewards and $R\colon S \times A \times S \to \mathbb{R}$ is the immediate reward function. The reinforcement learning agent needs to learn a policy, i.e. a mapping between states and actions that maximizes the received rewards.

The transition function T in an MDP is characterized by the Markov property. This means that the current state is only dependent on the previous state-action combination, and not on those before the previous time step. It is only feasible to solve an MDP when all state variables are observable. In the real world this is not the case. An agent that controls a heat pump will never have access to the temperature of the walls in the house it provides heating to. This is an important feature however, as the walls transfer heat from the outside world to the building's interior and vice versa. Such a setting would therefore be characterized by a Partially Observable MDP (POMDP) [11] where the Markov property does not hold. The learning agent can use a history of states to acquire a better approximation of the underlying state s_t and recover the Markov property [14].

2.2 Fitted Q-iteration

FQI is a batch reinforcement learning algorithm for value iteration that was first introduced by Ernst et al. [12]. Riedmiller et al. later proposed to use neural networks to approximate the Q-function [21]. The algorithm gathers experience online according to a preset policy and then engages in an offline training loop. Within the loop, targets are calculated according to equation 1, which uses the current estimation of the Q-value of the next state. The second part of the loop entails the learning of a mapping from state-action pairs $S \times A$ to the Q-value targets. If enough data is collected, the Q-value estimations will converge.

$$Q(s_t, a) = c(s_t, a, s_{t+1}) + \gamma \min_{a \in A} Q(s_{t+1}, a) \tag{1}$$

2.3 Multi-agent Reinforcement Learning

When transitioning to the multi-agent RL (MARL) scenario, we consider the case of independent learners (IL) [6] interacting in the same environment. The reward signal is usually an important aspect in a MARL system as it can determine the quality of the resulting collective behaviour. Two common choices are the *local reward* (L) providing local information to each individual agent depending on its own behaviour and the *global reward* (G) which expresses the total system utility and should align the system and agents' interests. A characteristic of our problem setting is the alignment between the local and global rewards, since minimizing the electricity consumption price for each device leads to a global minimization of the household's energy bill. Nevertheless, in order to allow the agents to coordinate and adhere to a maximum capacity constraint, a richer reward signal is necessary.

Difference reward (D) [30] is a reward signal that provides each agent with its own individual contribution to the total reward of the system, solving the credit assignment problem. D filters the noise in the reward signal attributed to the presence and actions of other agents. Given a global system utility G, the difference reward for agent i is expressed as:

$$D_i(z) = G(z) - G(z_{-i}) \tag{2}$$

where z denotes a general term for either state, action or state-action pair, according to the considered domain, and $G(z_{-i})$ is the global utility of a system from which the contribution of agent i is eliminated.

3 METHODS

3.1 Problem Setting

The problem setting in this paper is one where multiple devices, with both interruptible and uninterruptible loads, are managed by a demand response application. Every time step t, an agent has to decide whether to turn a device on or to let it remain idle. The duration of a time step is set to 15 minutes with $t \in \{1 \ldots 96\}$, as is usually done in similar smart grid use cases [4, 5, 7, 10, 23–25, 28]. The optimization horizon H of each agent is therefore 96 time steps or one day. At the end of the day, each agent receives the same global day-ahead electricity market price λ. In order to shift the electric loads to low-price periods, the agents controlling the devices are tasked with a cost-minimizing objective.

The devices simulated in this paper are a HP, EWH and DW. Each device is expected to minimize costs, while executing the following tasks:

1) Heat Pump: The HP maintains the interior air temperature T_t^{air} of a building between two temperature constraints T^{min} and T^{max}. These are set to 19°C and 23°C respectively, while the rated power of the device is set to 3 kW. These parameters were chosen in accordance to [23]. Due to the thermal inertia of the building, the walls function as thermal storage. The heat stored in these walls is dependent on both T_t^{air} and the outside air temperature T_t^{out}. As T_t^{air} goes down, the heat stored will be released back into the area to slow down the cooling process.

2) Electric Water Heater: The EWH in this setting has a rated power of 2.4 kW and a water buffer with a volume of 160 litres. The water contained in an EWH is stratified into different layers. The temperature of the top layer T_t^{out} is maintained between two temperature constraints of 45°C, T^{min}, and 60°C, T^{max}, similar to [10]. A time series of real tap water demand is included in the simulation. When an amount of water m_t is drawn from the tank, heat is extracted proportional to m_t.

3) Dishwasher: The DW is assumed to be fully loaded at the start of every day and needs to be turned on once a day. If the agent fails to attain this goal, it will incur a penalty.

Two agent designs are considered for this problem setting. One design includes a central agent which controls all devices at once. The second design, IL, trains one agent per device, where difference reward D is used to coordinate the agents. Agents have to take into account device-specific constraints, which are enforced by device-specific backup controllers. This paper illustrates that the FQI implementation can be beneficial for a multi-agent system when combined with difference reward D.

3.2 Markov Decision Process

3.2.1 State space. Each device has its own state s. In the case of IL, each agent has access to the state of the device it controls. For

	s_t
EWH	$t, T_t^{output}, m_t, a_{t-1}^{phys}$
HP	$T_t^{air}, T_t^{out}, a_{t-1}^{phys}$
DW	a_{t-1}^{phys}

Table 1: Observed state for every device.

the centralized setting, these states are concatenated into one larger state space. Table 1 describes the states of the different devices. The EWH's state contains the output temperature T_t^{output} of the water in the buffer, which is the water temperature at the output valve of the EWH. Time step t and tap water demand m_t are included so that the agent might learn a time-variant pattern. All three devices include the last physical action a_{t-1}^{phys} that was executed. The HP adds to that the interior air temperature T_t^{air} and the outside temperature T_t^{out}.

3.2.2 Action space and backup controller. The interruptible loads, the HP and EWH, each have their device-specific backup controller. This controller takes as input the agent's chosen action, evaluates the constraints and overrides the chosen action a_t with a_t^{phys} if the constraints are violated. Equation 3 formalizes this procedure. For the HP and EWH, T equals T_t^{air} and T_t^{output} respectively. Both a_t and a_t^{phys} have an action space $A = \{0, 1\}$, in which 0 corresponds to the idle action, and 1 corresponds to employing the full electric power potential of the device to heat the corresponding medium.

$$\mathcal{B}(s_t, a_t) = \begin{cases} a_t^{phys} = 1, \text{if } T < T^{min} \\ a_t^{phys} = 0, \text{if } T > T^{max} \\ a_t^{phys} = a_t \end{cases} \tag{3}$$

3.2.3 Cost function. Every physical action a_t^{phys} leads to a certain amount of energy ΔE that is used during time step t, depending on the energy consumption of each device $d \in \mathcal{D}$:

$$\Delta E_t = \sum_{d \in \mathcal{D}} \Delta E_t^d \tag{4}$$

The cost of this energy is described by equation 5. λ is the day-ahead market price of electricity, which is variable throughout the day and is communicated to the agents before training. This cost serves as an input to the Bellman update presented in equation 1.

$$c(s_t, a_t, s_{t+1}) = \Delta E_t \lambda_t \tag{5}$$

As a consequence, optimizing on equation 5 will result in all devices turning on at the same time during the lowest price-periods. This will lead to consumption peaks, thus straining the grid. To avoid this scenario, a constraint will be added stating that, at all times, the household's total energy consumption should be below a specific threshold τ. Otherwise, the devices incur a penalty. The penalty is proportional to the amount by which the threshold is exceeded. In this case, if $\Delta E_t > \tau$, we use $\Delta \hat{E}_t$ instead for the consumption in equation 5, computed as:

$$\Delta \hat{E}_t = (\Delta E_t - \tau)M + \Delta E_t \tag{6}$$

where τ is the threshold, and M a multiplier weighing the importance of the penalty. Note that the penalty is applied on the energy consumption, before multiplying with the price. The centralized learner will use equation 6 to learn that not all devices are allowed to be turned on at the same time.

Independent learners need a way to assign credit to the different agents. The difference rewards in equation 8 solve this credit assignment problem. IL should anti-coordinate thanks to the difference rewards, which means that during the same low price period one device should turn on a little bit earlier (or later) than the other devices so that they do not violate the grid constraint collectively. The device i's consumption when penalized (i.e., when $\Delta E_t > \tau$) will be different depending on the value of G_{-i}, calculated as:

$$G_{-i} = \sum_{\substack{d \in \mathcal{D} \\ d \neq i}} \Delta E_t^d \tag{7}$$

$$D_i = \Delta \hat{E}_t^i = \begin{cases} \Delta E_t^i M + \Delta E_t^i & , G_{-i} > \tau \\ (\Delta E_t - \tau)M + \Delta E_t^i & , G_{-i} \leq \tau \end{cases} \tag{8}$$

Thus, devices that are not involved in overloading the grid (such as the DW in our experiments) will receive a lower penalty than the ones that are. The goal of these difference rewards is to have the IL anti-coordinate.

When the total device consumption does not exceed the threshold (i.e., when $\Delta E_t \leq \tau$), then the difference rewards are equivalent to the local consumption of each device i: $D_i = \Delta E_t^i$. The final cost for each device i will be computed using D_i in equation 5.

3.3 Implementation

Algorithm 1 shows the implementation details of FQI for the application that is discussed in this paper. First, experience tuples $(s_k, a_k, s_{k+1}, \Delta E_k)$ are collected. ΔE_k is the electric energy consumed during the time step at which that tuple was experienced. These tuples are saved in a data batch $\mathcal{F}$ and used as input to the algorithm along with a day-ahead market price λ. Training is postponed until the end of an episode, which is a day in this paper. During the training procedure, an approximator $\hat{Q}_t$ fits the experience following a receding horizon approach (line 3) [5]. This implementation of FQI uses one $\hat{Q}_t$ per time step t. Therefore there are 96 approximators in this algorithm. For every tuple k in $\mathcal{F}$, Q-value targets Q_k are recomputed using the day-ahead electricity cost λ_t, the electric energy ΔE_k, and $\hat{Q}_{t+1}$, which is the approximator for the next time step (line 5 and 6). When applying the Bellman equation in this way, Q_t will be updated using the estimates of Q_{t+1} (except $\hat{Q}_{96}$ which will only receive the final cost).

Each approximator was trained with a small fully connected neural network, using Adam [16] as the optimizer. In order to improve stability, costs were normalized. The Q-update is adjusted according to the following scheme:

$$Q_t \leftarrow \frac{1}{H - t}(c_k + (H - t + 1) \min_{a \in A} \hat{Q}_{t+1}(s_{t+1}, a_t)) \tag{9}$$

where $H = 96$ is the horizon. As c_k is normalized, $Q_t \in [0, 1]$ at every time step, while still approximating the expected return. Note that, as the agent's goal is to minimize the total cost, equation 9 will use the smallest estimate $\hat{Q}_{t+1}$.

Algorithm 1 Fitted Q-iteration

Inputs: $\mathcal{F} = \left\{s_k, a_k, s_{k+1}, \Delta E_k\right\}_{k=1}^{\#\mathcal{F}}, \lambda$
$\forall\, s_k \in S, a_k \in A : \hat{Q}_H = 0$
for $t = H-1, ..., 1$ **do**
  for $k = 1, ..., \#\mathcal{F}$ **do**
    $c_k \leftarrow \lambda_t \Delta E_k$
    $Q_k \leftarrow c_k + \min_{a \in A} \hat{Q}_{t+1}(s_{k+1}, a)$
  end for
  use approximator to obtain $\hat{Q}_t$ from $\mathcal{TS} = \left\{(\left[s_t, a_t\right], Q_k), k = 1, ..., \#\mathcal{F}\right\}$
end for

In the case of the independent learners, the above procedure cannot be implemented in a straightforward way, as the issue of credit assignment might arise. This is where difference rewards start playing an important role. In the unconstrained case, the difference reward each agent receives is equivalent to the agent's local reward (i.e., the negation of the cost incurred by that respective device through its electricity consumption). Difference rewards show their true value in the constrained case, when the agents violate the grid constraint collectively and receive a penalty, as it allows every agent to only receive the portion of the penalty for which it is responsible.

4 EXPERIMENTS

In all simulations, the energy consumption of a household (composed of a HP, EWH and DW) is simulated for 40 days. At the end of each day the agent receives the day-ahead price profile λ. The agents use FQI to learn a control policy that will optimize the cost for the next day based on λ. The policy used by each agent is ϵ-greedy, with $\epsilon = 1$ during the first day and a decay of 0.1 with each ensuing day. This means that from day 10 onwards the policy is fully greedy. The discount factor γ is 1.

Q-values are approximated with one fully connected neural network for each timestep, and are optimized using Adam, with a learning rate of 0.01, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Each net is trained for maximum 3000 epochs. The training is interrupted if the loss does not improve for 100 consecutive epochs. Additionally, each net is fed with a history of four consecutive states in order to make up for the partial observability of the environment, as discussed in section 2.1. Because this technique recovers the Markov property, the problem setting is explicitly modelled as an MDP in section 3.2, and not as a POMDP.

4.1 Unconstrained learning

In this first simulation, there is no maximum consumption constraint imposed on the household. Validation of the correct workings of the most basic setup is thus provided here. The centralized learner receives the joint state of all individual devices and decides for each device which action it should take. For this setting, the 96 neural networks are composed of two fully connected hidden layers of five neurons each. In the IL setting, each device is equipped with an independent FQI learner which uses its own experience. The neural nets are composed of one hidden layer with eight neurons.

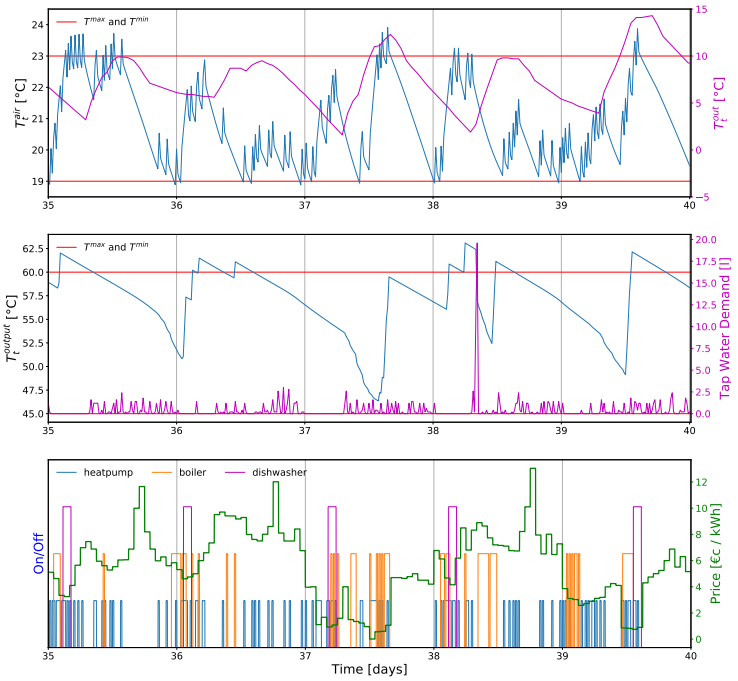

Figure 1: Detailed consumption of independent learners over a span of 5 days. Top: HP. Middle: EWH. Bottom: Actions executed by the respective devices.

Figure 1 shows an example of the behaviour of the independent learners during the last five days of the simulation. The top chart shows the inside temperature T_t^{air} (blue), bound between the comfort settings (T^{min} and T^{max} in red), and the outside temperature T_t^{out} (purple). The middle chart shows the output temperature of the EWH T_t^{output} (blue), also bound between its respective T^{min} and T^{max} (red), and the tap water demand m_t (purple). The bottom chart depicts the price (green) at each time step, as well as the timing at which devices are turned on or off. These are plotted with different heights to improve readability.

Due to the nature of the day-ahead market setting, the agents are optimising the cost on a per-day basis. The consequences of this are visible in figure 1, especially in the top chart (HP). Systematically, the agent controls T_t^{air} to equal T^{min} at the end of every day, since there is no benefit in spending more energy to heat the room. Consequently, the next day, the HP agent will have to heat the room irrespective of the electricity price. This has a smaller impact on the EWH, as this device is built specifically to have a larger thermal capacity.

Peaks in electricity price are usually avoided as can be seen in the bottom chart of figure 1. On day 38, we notice that the EWH is forced to turn on during a high price period, due to the sudden peak in water consumption. The DW also learned to turn on once a day on low price periods, while taking into account the uninterruptibility property of a cycle.

The performance of independent and centralized learners is compared in figure 2. The chart shows the cumulative electricity

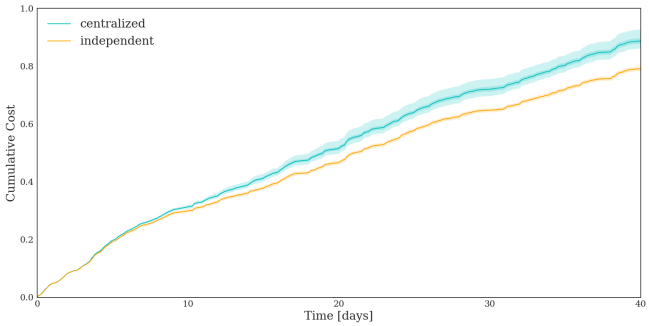

Figure 2: Cumulative cost for forty days with unconstrained agents. The chart represents an average over 21 runs per agent design. IL achieve 9.65% lower cost on average compared to the centralized agent after forty days.

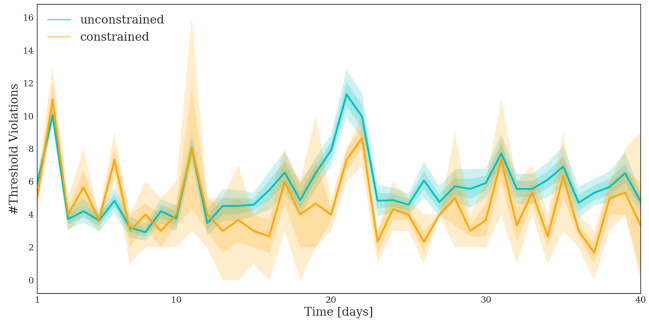

Figure 3: Number of times the threshold was exceeded per day, for centralized learners.

cost over 40 days. During the first few days, there is virtually no difference in cost due to the high exploration rate. However, starting from day 10, independent learners clearly manage their devices better than the centralized learner. This is to be expected, as the centralized learner's state and action space combines the states and actions of all household devices. Therefore, it has a bigger parameter space to search through to find a good approximation of the Q-function.

4.2 Constrained learning

The previous section focused entirely on minimizing the cost. However, this means the devices will often be turned on together when the price is low, thus straining the electricity grid. In order to spread energy consumption over the day, the previous experiment is extended with an additional constraint: minimizing the cost while satisfying a grid constraint $\Delta E_k < \tau$.

For the experiments in this section, a threshold of $\tau = 4 \times 10^6$ Ws is chosen. This corresponds to an electric energy usage of 1.11 kWh during one time step, which serves as a proxy for the capacity constraint on the feeder. τ is exceeded whenever the HP and EWH are turned on at the same time, but not when either of these devices are turned on together with the DW. A multiplier $M = 4$ is chosen to penalize constraint violations. The other settings are kept equal to the unconstrained simulations.

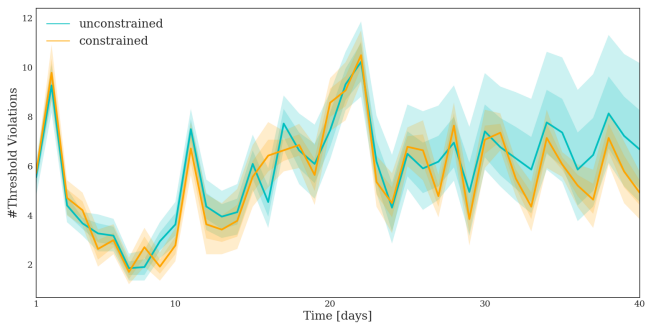

Figure 4: Number of times the threshold was exceeded per day, for independent learners.

Figure 3 shows the number of times threshold τ was exceeded during each day by the centralized learner in both the unconstrained and constrained case. From day 10 onwards, when the agents are operating fully greedily, the constrained learners systematically exceed the threshold less often than the unconstrained ones. This indicates that the penalty is taken into account. However, the variability is often larger in the constrained case. Nevertheless, the constrained learner exceeds the threshold every single day, and is unable to fully anti-coordinate its devices.

Domain-specific dynamics play a significant role in this demand response setting. While the change in outside temperature is quite smooth through time, tap water demand varies rapidly and is quite unpredictable. Therefore, it might be easier to anticipate the HP's usage compared to the usage of the EWH. This is especially noticeable on day 20, where an unusually large amount of water is requested, forcing the EWH to be active immediately afterwards for a duration of 2 to 5 hours. This results in a consistently high number of threshold violations.

The same results for IL are visualized in figure 4. The constrained independent learners use the same parameter settings as the unconstrained ones. $\tau = 4 \times 10^6$ and $M = 4$. Agents should anti-coordinate thanks to the difference rewards implementation as seen in equation 8, but IL with constraints clearly does not outperform IL without constraints. Additionally, the centralized learner from figure 3 attains lower threshold violation counts. In the final days of the simulation in figure 4 the constrained learner achieves lower threshold violations on average, but further research is necessary to develop learners which can minimize the amount of violations effectively.

4.3 Discussion and Future Work

Section 4.1 shows the benefit of having one learner per device in the demand response problem presented in this paper. IL achieve on average an electricity cost that is 9.65% lower than in the centralized case and are thus more effective in shifting electric load towards low-price periods. However, when confronted with an extra constraint on the level of the house, IL experience more difficulty to satisfy this constraint. Both agent designs fail to keep energy usage below the threshold, but show promising preliminary results as the constrained learners have a lower count of threshold violations in the long term.

Future work will concentrate on developing adequate learners for this problem in a number of ways. Firstly, it might be worth looking at this setting as a Multi-objective Reinforcement Learning (MORL) [22] problem, on which a naïve linear scalarization is applied. Methods such as [2, 13, 19] were previously succesfully implemented in combination with FQI [3]. More explicit cooperation and coordination mechanisms between the agents could help in more effectively aligning when to turn on or stay idle, rather than the implicit approach through anti-coordination in this paper. An example would be to use Coordinating Q-Learning [8, 9], where the agents mostly act independently, but learn on which states they should take the other agents into consideration. Additionally, auxiliary tasks [15] could be taken into account, such as predicting the tap water demand, to investigate if learning on such secondary pseudo-rewards can help the agent(s) attain a better performance. Finally, the price profile is implicitly provided since it is used in the calculation of the targets for the different Q-networks. This information could be explicitly provided by including λ in the state space of the agent, so that it can directly learn how the price is related to the immediate cost c_k and the expected cost of the rest of the trajectory.

5 RELATED WORK

Reinforcement learning has been used many times over to tackle Demand Response problems in previous research. In [29], Q-learning is used in order to schedule devices according to consumer choices and time preferences. Sancho-Tomás et al. [26] extended No-MASS to include demand response strategies, and used Q-learning to optimize load-shifting and charging and discharging of batteries. In [10], model-based RL was used to maximize self-consumption of local photovoltaic production, in the case of a domestic hot-water buffer. Taking the service provider's perspective, [17] uses Q-learning in a dynamic price demand response setting, in order to adaptively decide the retail price of energy.

FQI specifically matured in recent research as a valuable technique when tackling price-based demand response problems. Both clusters of devices and individual appliances were considered. In [10, 23, 28], thermostatically controlled loads are controlled as single devices in a day-ahead market setting. Ruelens et al. include forecasts of exogenous features and a model-free Monte Carlo method in combination with the FQI algorithm to achieve better cost performance compared to the naive FQI approach [23]. Costanzo et al. use model-assisted FQI and policy shaping to attain similar cost improvements [7]. Other work proves even feature engineering [4] can help extract critical information from the environment's state.

6 CONCLUSIONS

In this work FQI learners were implemented to optimize the cost of the energy consumption of a household with three devices in a day-ahead market price setting. Two approaches were illustrated for this application: a centralized single agent controlling all devices versus a multi-agent RL based-approach with independent learners that each control one device. In both settings, the agents are able to cope with the mix of interruptible and uninterruptible loads and optimize multiple devices concurrently. Due to the aligned common goals of all agents, fully independent learners are able to learn more cost-effective policies than a centralized agent and achieve better performance. On average, the independent learners attain an electricity cost that is 9.65% lower than the cost attained with the centralized learner.

In an attempt to avoid straining the electricity grid, a soft constraint was added to limit the amount of energy that is allowed to be consumed during one time step. Once this limit is exceeded, the agent(s) receives a penalty proportional to the violation. Although both centralized and independent learners show promising preliminary results, future work will look into more advanced techniques to coordinate multiple agents in this MORL setting. Previous work mentioned in this paper has shown promising results when combining FQI with other multi-objective techniques. Coordination schemes such as coordinated Q-learning, the inclusion of auxiliary tasks and the availability of additional information, such as the price profile, to the agent(s) are valuable paths to be considered in solving this research problem.

ACKNOWLEDGMENTS

This work is supported by Flanders Innovation & Entrepreneurship (VLAIO), SBO project 140047: Stable MultI-agent LEarnIng for neTworks (SMILE-IT).

REFERENCES

[1] M. H. Albadi and E. F. El-Saadany. 2007. Demand Response in Electricity Markets: An Overview. In *2007 IEEE Power Engineering Society General Meeting*.
[2] Andrea Castelletti, Giorgio Corani, A Rizzolli, R Soncinie-Sessa, and Enrico Weber. 2002. Reinforcement learning in the operational management of a water system. In *IFAC workshop on modeling and control in environmental issues*.
[3] A Castelletti, S Galelli, M Restelli, and R Soncini-Sessa. 2010. Tree-based reinforcement learning for optimal water reservoir operation. *Water Resources Research* (2010).
[4] Bert J. Claessens, Dirk Vanhoudt, Johan Desmedt, and Frederik Ruelens. 2018. Model-Free Control of Thermostatically Controlled Loads Connected to a District Heating Network. *Energy and Buildings* (2018).
[5] Bert J Claessens, Peter Vrancx, and Frederik Ruelens. 2016. Convolutional neural networks for automatic state-time feature extraction in reinforcement learning applied to residential load control. *IEEE Transactions on Smart Grid* (2016).
[6] Caroline Claus and Craig Boutilier. 1998. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI* 1998 (1998), 746–752.
[7] Giuseppe Tommaso Costanzo, Sandro Iacovella, Frederik Ruelens, Tim Leurs, and Bert Claessens. 2016. Experimental analysis of data-driven control for a building heating system. *Sustainable Energy, Grids and Networks* (2016).
[8] YM De Hauwere, P Vrancx, and A Nowé. 2011. Detecting and solving future multi-agent interactions. In *Proceedings of the AAMAS Workshop on Adaptive and Learning Agents, Taipei, Taiwan*. 45–52.
[9] Yann-Michaël De Hauwere, Peter Vrancx, and Ann Nowé. 2010. Learning multi-agent state space representations. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 715–722.
[10] Oscar De Somer, Ana Soares, Koen Vanthournout, Fred Spiessens, Tristan Kuijpers, and Koen Vossen. 2017. Using reinforcement learning for demand response of domestic hot water buffers: A real-life demonstration. In *Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2017 IEEE PES*.
[11] Alain Dutech and Bruno Scherrer. 2013. Partially observable Markov decision processes. *Markov Decision Processes in Artificial Intelligence* (2013).
[12] Damien Ernst, Pierre Geurts, and Louis Wehenkel. 2005. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research* (2005).
[13] Ying Guo, Astrid Zeman, and Rongxin Li. 2009. A reinforcement learning approach to setting multi-objective goals for energy demand management. *International Journal of Agent Technologies and Systems (IJATS)* 1, 2 (2009), 55–70.
[14] Matthew Hausknecht and Peter Stone. 2015. Deep recurrent Q-learning for partially observable MDPs. *CoRR* (2015).
[15] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. 2016. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397* (2016).
[16] D Kinga and J Ba Adam. 2015. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

[17] Renzhi Lu, Seung Ho Hong, and Xiongfeng Zhang. 2018. A Dynamic pricing demand response algorithm for smart grid: Reinforcement learning approach. *Applied Energy* 220 (2018), 220–230.

[18] Yael Parag and Benjamin K. Sovacool. 2016. Electricity market design for the prosumer era. *Nature Energy* (2016).

[19] Julien Perez, Cécile Germain-Renaud, Balázs Kégl, and Charles Loomis. 2009. Responsive elastic computing. In *Proceedings of the 6th international conference industry session on Grids meets autonomic computing*.

[20] Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (1st ed.). John Wiley & Sons, Inc.

[21] Martin Riedmiller. 2005. Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*.

[22] Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. A Survey of Multi-Objective Sequential Decision-Making. *Journal of Artificial Intelligence Research* 48 (2013), 67–113.

[23] Frederik Ruelens, Bert J. Claessens, Stijn Vandael, Bart De Schutter, Robert Babuška, and Ronnie Belmans. 2016. Residential demand response of thermostatically controlled loads using batch reinforcement learning. *IEEE Transactions on Smart Grid* (2016).

[24] Frederik Ruelens, Bert J. Claessens, Stijn Vandael, Sandro Iacovella, Pieter Vingerhoets, and Ronnie Belmans. 2014. Demand response of a heterogeneous cluster of electric water heaters using batch reinforcement learning. In *Power Systems Computation Conference*.

[25] Frederik Ruelens, Bert J. Claessens, Peter Vrancx, Fred Spiessens, and Geert Deconinck. 2017. Direct Load Control of Thermostatically Controlled Loads Based on Sparse Observations Using Deep Reinforcement Learning. *arXiv preprint arXiv:1707.08553* (2017).

[26] A Sancho-Tomás, J Chapman, M Sumner, and Darren R. 2017. Extending No-MASS: Multi-Agent Stochastic Simulation for Demand Response of residential appliances. In *Proceedings of the Building Simulation Conference*.

[27] Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*. MIT press Cambridge.

[28] K. Vanthournout, R. D'hulst, D. Geysen, and G. Jacobs. 2012. A Smart Domestic Hot Water Buffer. *IEEE Transactions on Smart Grid* (2012).

[29] Zheng Wen, Daniel O'Neill, and Hamid Maei. 2015. Optimal demand response using device-based reinforcement learning. *IEEE Transactions on Smart Grid* (2015).

[30] David H Wolpert and Kagan Tumer. 2001. Optimal payoff functions for members of collectives. *Advances in Complex Systems* 4, 2/3 (2001), 265–279.