ADAPTIVE MULTI-AGENT CONTROL OF HVAC SYSTEMS FOR RESIDENTIAL DEMAND RESPONSE USING BATCH REINFORCEMENT LEARNING

José Vázquez-Canteli[1], Stepan Ulyanin[2], Jérôme Kämpf[3], Zoltán Nagy[1]

[1]Intelligent Environments Laboratory, Department of Civil, Architectural and Environmental Engineering, The University of Texas at Austin, Austin, TX, USA

[2]School of Computer Science, Georgia Institute of Technology, Atlanta, GA,

[3]Haute Ecole d'Ingénierie et d'Architecture Fribourg, Fribourg, Switzerland

ABSTRACT

Demand response allows consumers to reduce their electrical consumption during periods of peak energy use. This reduces the peaks of electrical demand, and, consequently, the wholesale electricity prices. However, buildings must coordinate with each other to avoid delaying their electricity consumption simultaneously, which would create new, delayed peaks of electrical demand. In this work, we examine this coordination using batch reinforcement learning (BRL). BRL does not require a model, and allows the buildings to adapt over time to the optimal behavior. We implemented our controller in CitySim, a building simulator, using TensorFlow, a machine learning library.

INTRODUCTION

Residential buildings account for about 30% of the global energy consumption, of which space conditioning constitutes a large portion (IEA 2013). In regions where summers can be very hot, air conditioning can produce high peaks of electrical demand which lead to constraints in the power transmission lines that can cause very high wholesale prices of electricity (Dupont, De Jonghe, et al. 2014). Furthermore, in developing countries, the demand for air conditioning devices if expected to increase significantly in the coming years (McNeil and Letschert 2008).

Distributed renewable energy resources, such as photovoltaic PV panels, can improve the energy autonomy of residential consumers, reduce CO_2 emissions, and reduce the peaks of electrical demand of the power grid. However, high penetration of renewable energy resources can cause instability problems in the electrical infrastructure because of their limited predictability, controlability, and high variability (Dupont, Dietrich, et al. 2014).

Demand response can enable consumers to reduce their electrical consumption during periods of peak energy demand in exchange for a lower energy bill (Siano 2014). Furthermore, demand response can improve grid stability by increasing demand flexibility, and by shifting peak demand towards periods of peak renewable energy generation, if available. Two examples of implementations of demand response programs in the U.S. are the Energy-Smart Pricing Plan℠ in Illinois from 2003-2006 (Summit Blue Consulting 2007), and the Critical Peak Pricing experiment in California (Herter et al. 2007). However, when multiple buildings delay their electrical consumption simultaneously, they can produce new peaks when prices of electricity were expected to be lower. To avoid such rebound effects, buildings must provide a coordinated response, which can be either cooperative or competitive (as it is the case in this paper).

Advanced control approaches, such as Model-Predictive Control (MPC) (Rault 1978), can achieve near-optimal energy cost savings in systems for which a mathematical model is available. However, such models are often too time and cost intensive to implement in medium sized residential buildings (Shaikh et al. 2014). Moreover, if buildings are retrofitted, their models are no longer be accurate.

Reinforcement learning (RL) is a model-free learning algorithm that can adapt to changing factors such as weather conditions, building retrofitting, or the installation of additional solar PV capacity. RL does not require any kind of model identification, but rather behaves as a "plug and play" controller. It can learn both on-line (as it takes control actions), and off-line (from historical data, or by observing another controller). Off-line learning is particularly useful because it allows RL to take advantage of the growing amount of sensor data that there will be available for buildings. It also allows any RL controller to be coupled with a secondary or "back-up" controller from which it can learn. Furthermore, if the RL controller has not learned enough to take the appropriate control decisions, it can switch to the back-up controller and keep learning from the sensor data.

Reinforcement learning was used for the first time in the built environment by Mozer in his neural network house project in 1998 (Mozer 1998). Since then, some research has focused on the use of RL to minimize the cost of electricity in buildings with energy storage devices and renewable energy resources (Ruelens et al. 2014), maximize the self-consumption of local PV generation by storing the energy in DHW buffers (De Somer et al. 2017), or control a building energy system with several photovoltaic-thermal panels, geothermal boreholes, (Yang, L., et al. 2015). However, little research has been done in the use of RL to coordinate different buildings sharing the same electricity prices, which are dependent on their cumulated energy demand. RL has been used to coordinate several HVAC systems in a double-auction market using GridLAB as the simulation environment (Sun et al. 2015). However, the researchers focused on modifying the indoor temperature of the buildings to allow for more discomfort when the price of electricity was higher, which in a real-world scenario could discourage consumer from participating in demand response programs. The implementation of multi-agent reinforcement learning in the field of demand response still needs further research in order to achieve scalable control systems that allow buildings to learn from each other (Vázquez-Canteli, J.R., and Nagy 2018).

Previous research (Vázquez-Canteli, J.R., Kämpf, J. H., and Nagy 2017) demonstrated how batch reinforcement learning (BRL) can achieve significant energy savings in a single building with a heat pump and a chilled water tank. In this paper, we demonstrate how BRL can be used to reduce the cost of the electricity consumed by multiple buildings in a demand response scenario. Another important contribution is that we created a new simulation environment that allows us to use a building energy simulator designed for urban scale analysis, CitySim (Robinson 2011), combined with a powerful machine learning library, TensorFlow (Agarwal et al. 2015), that allows us to take advantage of advanced machine learning algorithms. Finally, we test this simulation environment with a case study of two residencial buildings located in Austin, TX. In the case study, we simulated two buildings that share the same prices for electricity. In each building, a heat pump provides the necessary cooling, and a water tank provides storage capabilities. The price for electricity increases with the electrical demand of both buildings. Therefore, the BRL controller of each building must learn how to compete with the other building to achieve greater cost savings and avoid consuming energy simultaneously. We also demonstrate how BRL can adapt to the installations of PV panels on top of the buildings, which completely changes the dynamics of the system.

METHODOLOGY

Reinforcement learning

Reinforcement learning can be formalized using a Markov Decision Process (MDP), which contains four elements: a set of states S, a set of actions A, a reward function R: S x A, and transition probabilities between the states P: S x A x S ϵ [0,1]. A policy π then maps states to actions as π: S→ A, and the value function $V^{\pi}(s)$ of a state s, given by the Bellman equation, eq. 1.

$$V^{\pi}(s) = r(s, \pi(s)) + \gamma \sum P(s, \pi(s), s')\, V^{\pi}(s') \tag{1}$$

is the expected return for the agent when starting in the state s and following the policy π. In (1) r is the reward received for taking the action $a = \pi(s_k)$, and $\gamma \,\epsilon$ [0,1] is a discount factor for future rewards. The goal of the agent is to find a policy that maximizes its rewards. An agent that uses $\gamma = 1$ will focus on long term rewards, whereas an agent using $\gamma = 0$ will seek immediate rewards. RL is particularly useful when the model dynamics (P and R) are not known, and have to be determined or estimated through interaction of the agent with the environment as depicted in Fig. 1.

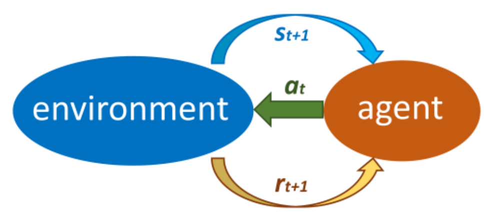

Figure 1 Agent and environment interaction

Q-learning is the most widely used algorithm to solve eq. 1 due to its simplicity (Watkins and Dayan 1992). In simple tasks with small finite state sets, all transitions can be represented with a table storing state–action values, or q-values. Each entry in the table represents a state–action (s,a) tuple, and Q-values are updated as

$$Q_{k+1} = Q_k + \alpha[r + \gamma \max Q - Q_k] \tag{2}$$

$\alpha \in (0,1)$ is the learning rate, which explicitly defines to what degree new knowledge overrides old knowledge: For $\alpha = 0$, no learning happens, while for $\alpha = 1$, all prior knowledge is lost at each iteration. It can be shown, that the optimal value for a state, $V^*(s)$, is given by

$$V^*(s) = \max_a Q^*(s,a) \tag{3}$$

Where $Q^*(s,a)$ is the optimal Q value for state s. The drawback of this tabular approach is that only one Q-value is updated at a time, and requires the states and actions to be discrete. When the state-action space is larger, and has continuous values, the Q-table is substituted by an Artificial Neural Network (ANN) (Busoniu et al. 2010) that maps states and actions directly to their Q-values, Fig. 2. This version of Q-learning that uses ANNs to estimate the Q-values from their states and actions is known as batch reinforcement learning (BRL) (Kalyanakrishnan et al. 2008). Table 1 shows the BRL algorithm work as implemented in this work.

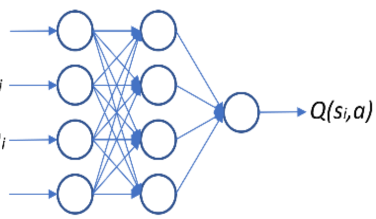

Figure 2 Artificial neural network for batch reinforcement learning

Table 1 Batch reinforcement learning

```
Q̃ ← Initialize ANN randomly
i ← 0
Repeat
    (s_i, s'_{i-1}) ← scale(s_t, s_t)
    a_i ← actionSelection(s_i, Q̃)
    s_{t+1} ← takeAction(s_i, a_i)
    r_i ← getReward(s_i, a_i, s_{t+1})
    D_i ← (s_i, a_i, r_i, s'_i)
    If (i + 1) % batchSize == 0
        Repeat
            a* ← arg max_a Q̃(s', a)
            Q(s, a) ← r(s, a) + γ Q̃(s', a*)
            Q̃ ← trainANN(s, a, Q)
        Until p == epochs
    end if
    If t % stepSize == 0
        i + +
    end if
until t = simulation time
```

To find the best action for a given state, BRL uses a trade-off between exploring actions that the algorithm is uncertain on how good they are (may not have high Q-values at that time) and exploiting actions that seem to provide high long-term rewards (have high Q-values).

Two of the most popular action-selection algorithms are ϵ-greedy, and soft-max action-selection. This work uses soft-max action-selection, which selects the actions with a probability that is related to the Q-value as

$$\Pr(a|s) = \frac{e^{\frac{Q(a,s)}{T}}}{\sum e^{\frac{Q(s,e)}{T}}} \tag{4}$$

Actions with higher Q-values are more likely to be selected than actions with lower Q-values. T is the Boltzmann temperature constant. High values of T (i.e. T > 3) make the probability of selecting any action very homogeneous regardless of their Q-values. Low values of T (i.e. T < 0.1) lead the action-selection algorithm towards a greedy policy, in which actions with the highest Q-values are selected most of the time. It is convenient to start the learning process with high values of T to increase exploration, and then reduce T to exploit the acquired knowledge to maximize the rewards obtained.

CitySim and TensorFlow

To perform the simulations, we created a framework that allows us to take advantage of the features of a building energy simulator for urban scale analysis, and advanced machine learning algorithms, (i.e. different kinds of artificial neural networks and training algorithms).

CitySim is a building energy simulator developed at École Polytechnique Fédérale de Lausanne (EPFL) that computes an hourly estimation of the energy demand for heating, cooling, and lighting in every building (Robinson 2011). CitySim is validated (Walter and Kämpf 2015), and has been used, e.g., in urban retrofit analysis (Vazquez-Canteli and Kampf 2016). It allows us to easily create scenarios with multiple buildings, make changes (i.e. building retrofitting, population growth, construction of new buildings, additional shadowing effects), and analyze how buildings can learn and adapt to those changes through reinforcement learning.

TensorFlow is an open source machine learning library created for efficient numerical computation, using data-flow graphs (Agarwal et al. 2015). Furthermore, there exist high-level APIs for implementation of machine learning algorithms, such as Keras, an open source library that uses TensorFlow as a back-end engine. Keras provided modular access to the backend features of TensorFlow that we used to build the neural network.

We implemented the reinforcement learning controller in CitySim, while we used TensorFlow to efficiently implement the artificial neural network that the controller needs. Fig. 3 depicts this framework we created for our simulations. This simulation framework is detailed in (Vázquez-Canteli, J.R., Ulyanin, S., Kämpf, J. H., Nagy 2018).

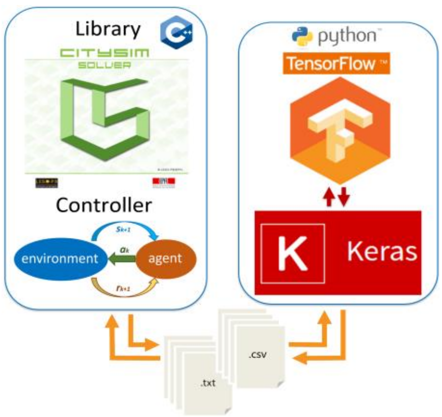

Figure 3 CitySim-TensorFlow framework

SIMULATION

We conducted our simulations for a case study under the climatic conditions of a typical year in Austin, TX. We chose a period of hot weather comprised of 122 days for our simulation, between May the 19th and September the 7th, as Fig. 4 illustrates. We selected this period because it was comprised between a relatively small and steady range of warm temperatures that would allow us to study the performance of our controller when implemented in a cooling system. The weather file was obtained from *Meteonorm*.

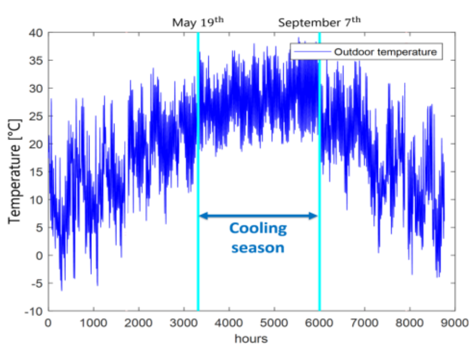

Figure 4 Selected period of cooling that we chose for the simulation.

Building energy models

The building energy system of this study is comprised of an air-source to water heat pump, that provides cooling energy to a chilled water tank, which stores the water and provides cooling to the building. The objective of the controller is to reduce the dependency of the heat pump on the electrical grid by storing and releasing cooling energy from the chilled water tank at different times.

A photovoltaic array may provide electricity to the heat pump, and we assume that the buildings must either store or consume the electricity they generate. The excess energy cannot be sold to other buildings. Therefore, if the PV panels have the capacity of generating more electricity that can be stored or consumed at a given time, such capacity is not utilized. During the day the PV panel reduces the electricity that the heat pump consumes from the grid, but the coefficient of performance (COP) of the heat pump will typically be lower during these hours. During the night, the heat pump will typically benefit from a higher COP, but will lack the electricity generated by the PV panel.

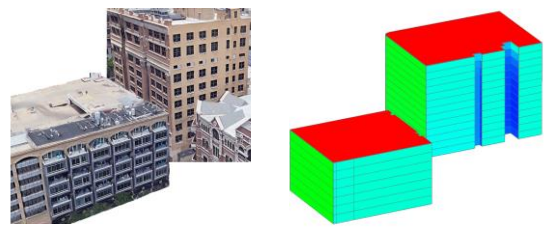

Figure 5 Building envelopes and the representation of their building energy models in CitySim

For the case study, we modelled one seven-story, and one nine-story residential building, both located in downtown Austin, TX. The buildings are illustrated in Fig. 5, and their physical characteristics have been estimated based on typical values that complied with the norm ASHRAE Fundamentals 90.1. We used infiltration rates of 0.9 h^{-1}, and 0.8 h^{-1} respectively, windows with an U-value of 2.135 W/m^2K for one building, and 6.23 W/m^2K for the other. The solar energy transmittance coefficients (G-value) of the windows were 0.49 and 0.62 respectively, and we assumed window to wall ratios of 0.3 for both buildings.

Table 2 Materials of construction of the envelopes

	MATERIAL	THICK. [m]	THERMAL. COND. [W/(mK)]	Cp [J/kgK]	DENSITY [kg/m^3]
WALLS	Rendering	0.02	0.87	1100	1800
	PS30 polystyrene	0.10	0.036	1400	30
	Reinfor. concrete	0.17	2.40	1000	2350
	Plaster	0.01	0.43	1000	1200
FLOOR	Reinfor. concrete	0.3	2.40	1000	2350
ROOF	Rendering	0.02	0.87	1100	1800
	PS30 polystyrene	0.10	0.036	1400	30
	Reinfor. concrete	0.17	2.40	1000	2350
	Plaster	0.01	0.43	1000	1200

Table 2 contains the materials we used to model the walls of the envelopes of both buildings. The purpose of this paper is not to accurately model the buildings but rather demonstrate that buildings with different thermal characteristics can use the same reinforcement learning controller, and learn to coordinate with each other in a competitive way.

Electricity price

The price of electricity is proportional to the sum of the electrical consumption of both buildings at any given time. This constitutes an incentive for the buildings not to consume electrical energy simultaneously. The relation between the price of electricity (in USD), and the sum of the electricity consumption of both buildings (in kWh) at any given time is modeled as

$$\boldsymbol{P = 3 \cdot 10^{-5} \cdot E + 0.045} \tag{5}$$

In a real scenario with dozens of buildings, the increase of their electrical demand produces increases in the wholesale prices for electricity, which lead to higher retail prices of electricity in the long term. Eq. 5 is based on a reasonable estimation of retail prices provided by Austin Energy. In this paper, we do not intend to simulate the electricity market, but to show that buildings can learn and adapt to constant changes in the prices of electricity caused by the actions of other buildings, learn from each other and compete for a lower energy bill.

The controller state-action space

The objective of the batch reinforcement learning controller is to minimzie the cost of the electricity consumed by the heat pump from the power grid. Therefore, the reward that the controller receives is the cost of electricity. As states and actions we chose all those variables that help in predicting the future reward of the system. Fig. 6 illustrates all the variables we chose because of their influence over the cost of the energy buildings consume.

The action of the RL controller is the target temperature of the chilled water tank (for the next time-step), while the states are defined as the current temperature of the water in the tank, the outdoor temperature (which is a predictor of the energy demand in the building as well as of the coefficient of performance (COP) of the heat pump), the hour of the day, and the price of electricity. Indoor temperatures are always maintained between the appropriate temperature set-points, and they are never increased to achieve greater cost savings at the expense of thermal comfort. Since indoor temperature is maintained constant most of the time, it is not used as a state. Both the electricity prices and the COP of the heat pump have an influence on the overall cost of the electricity.

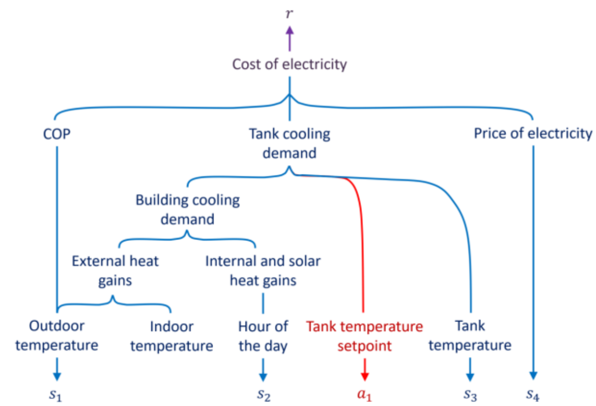

Figure 6 State-action space to predict the reward

Reinforcement Learning Controllers

We compared three different controllers: a rule-based controller (RBC), a single-agent BRL controller, and a multi-agent BRL controller. The RBC cools the water in the chilled water tank every time it reaches 20 °C until it reaches 10 °C. On the other hand, both the single-agent, and the multi-agent BRL controllers use reinforcement learning in each building to adjust the temperature of the tank every two hours.

The single-agent BRL controllers reward their respective buildings by calculating an energy cost that uses a virtual electricity price, which is calculated from eq. 5 using their individual electricity consumption as the input *E*. This virtual price is only used to calculate the reward for each controller, whereas the real price both buildings pay is computed using the sum of the electricity consumption of both buildings as the input *E* in eq. 5. This virtual price (at the previous time-step) was also used as the state "Price of electricity" that Fig. 6 shows.

The multi-agent BRL controllers reward their respective buildings using their real cost of electricity, which is calculated using the electricity price buildings share (using eq. 5 with the sum of both electrical demands as the input *E*). Therefore, the multi-agent BRL controller penalizes more the buildings if they consume electricity simultaneously, while the single-agent BRL controller only penalizes each of them, separately, for increasing their individual electricity consumption. The real price of electricity both buildings share (at the previous time-step) was also used as a state for every controller as Fig. 6 illustrates.

In a second experiment, we add a photovoltaic array on one of the buildings, covering 20% of its roof surface. We analyze how this can affect the electricity prices and whether the different controllers can adapt to this new situation.

RESULTS AND DISCUSSION

Fig. 7 shows how the single-agent BRL controller learned to cool the water tank when the outdoor temperature is low and the COP high, and discharge the cooling energy from the tank into the building when the outdoor temperature is high and the COP low. This control achieves greater energy cost savings than the RBC, which switches on and off without considering the outdoor temperature. On the other hand, the multi-agent BRL controller not only considers the outdoor temperature, but also the price of electricity both buildings share, which depends on the electrical consumption of the other building. Therefore, it did not follow a pattern intended to maximize the COP.

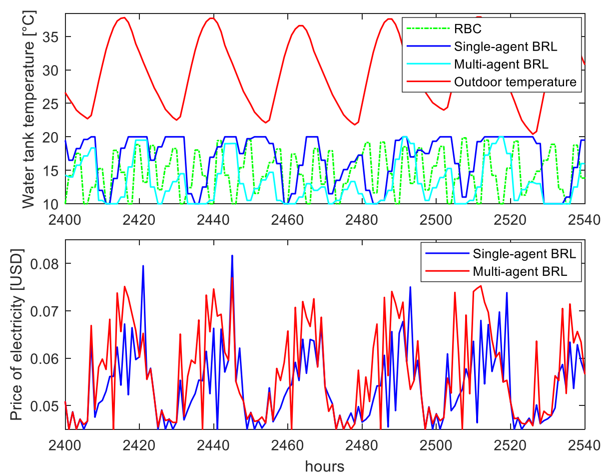

Figure 7 Variations of the temperature of the water tanks, with respect to the outdoor temperature, and electricity prices for one building and different controllers

Fig. 8 illustrates the energy cost of each building using the three different controllers. The cost of electricity is scaled for a better visualization of the cost reductions. The RLC led to the highest electricity costs in both buildings. The reason for this is that it is an on-off controller, which makes use of the heat pump at full power capacity when the water tank reaches 20 °C until it reaches 10 °C. This creates spikes in the energy consumption of both buildings, leading to high electricity prices and costs.

On the other hand, both the single-agent, and the multi-agent BRL controllers achieved the same improvement with respect to the RBC. However, the single-agent BRL controller optimized the COP of the heat pump, while the multi-agent BRL controller did not. Therefore, the multi-agent BRL controller achieved some level of coordination between the buildings to make sure that the best price was paid rather than the COP maximized. This shows that even though we violate the markovian property, we achieve similar results than the single-agent controller, which shows that coordination did happen.

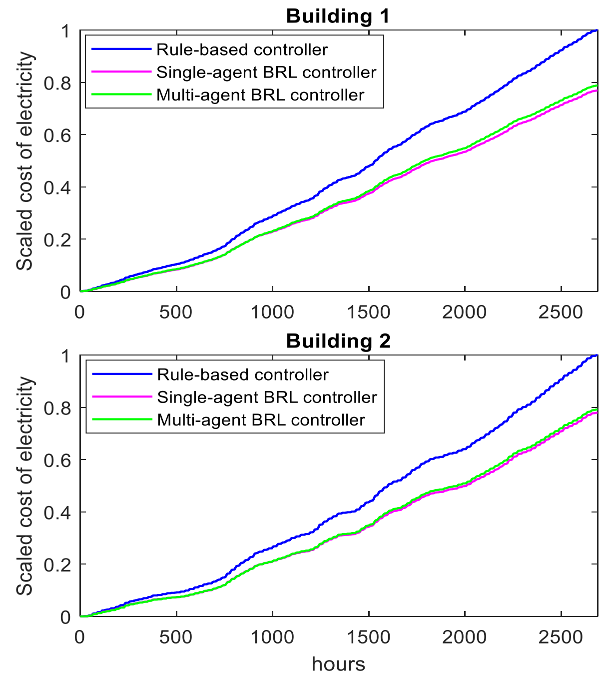

Figure 8 Scaled electricity cost of both buildings using three different controllers

Addition of a photovoltaic array

Fig. 9 illustrates the temperature of the tanks for the building that does not have a photovoltaic array, and the shared price of electricity. While the single-agent BRL controller still focused on COP maximization to increase the energy cost savings, the multi-agent BRL controller learned how, during the day, the other building generated electricity and reduced the price of electricity. Therefore, the multi-agent BRL controller tended to cool the water tank when the power output from the photovoltaic array of the other building was higher.

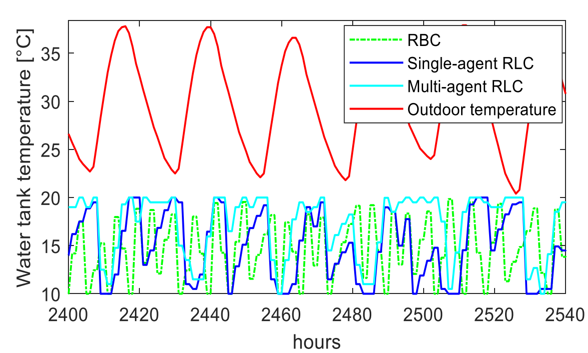

Figure 9 Variations of the temperature of the water tanks, with respect to the outdoor temperature

In this new scenario, as Fig. 10 shows, the multi-agent BRL controller of the building without PV panels achieves greater cost savings than the single-agent BRL controller. This is because it takes into consideration the effect that the photovoltaic generation of the other building has on the price of electricity.

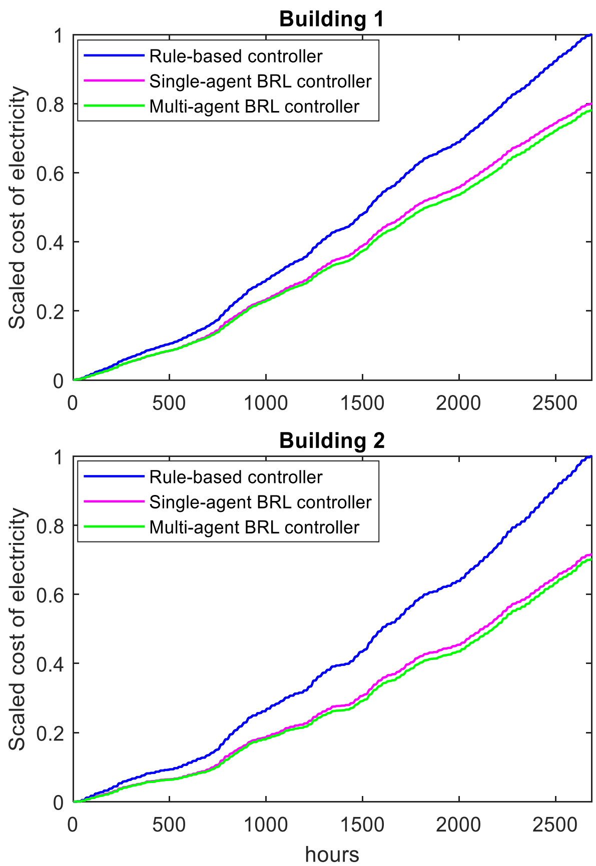

Figure 10 Scaled electricity cost of both buildings using three different controllers

CONCLUSION

Demand response allows consumers to reduce their electrical consumption during periods of peak energy use in exchange for a lower energy bill. This can help to reduce the peaks of energy demand, leading to lower wholesale prices of electricity. However, buildings must coordinate with each other to avoid delaying their energy consumption simultaneously, and creating new, delayed peaks of electrical demand.

We have used batch reinforcement learning (BRL), an adaptive algorithm that does not require model identification, to control the energy storage and supply in two buildings that buy electricity from the grid at price they both share. We have also created a new simulation environment that takes advantage of CitySim, a validated building energy simulator for urban scale analysis, and TensorFlow, a machine learning library. This simulation environment allows us to easily create scenarios with multiple buildings, and implement advanced machine learning algorithms efficiently.

We tested our batch reinforcement learning controller in a simulated case study. We showed how both, single-agent and multi-agent, BRL controllers achieved the same improvement in energy cost with respect to the RBC controller. However, the single-agent controller optimized the COP of the heat pump, while the multi-agent controller did not. Therefore, the multi-agent controller achieved some level of coordination between the buildings to make sure that the best price was paid rather than the COP maximized.

When a photovoltaic array was added on top of one of the buildings, the multi-agent BRL controller did achieve greater energy cost savings than the single-agent controller. The building without the PV array learned to store more cooling energy when the other building was generating a higher power output, and therefore, lowering the price of electricity.

Our further research will focus on investigating new multi-agent BRL frameworks, either competitive or cooperative, that will allow the buildings to share enough information more effectively to coordinate better with each other and achieve greater energy cost savings. We will try recurrent approaches, in which the future action of one controller is used a state for the other controller in an iterative way. Additionally, we will implement a model that represents electricity prices more realistically (e.g. an auction energy market). We will also account for variations of the temperature setpoints and the use of the thermal mass of the buildings to store additional energy.

REFERENCES

Agarwal, A. et al., 2015. TensorFlow : Large-Scale Machine Learning on Heterogeneous Distributed Systems.

Busoniu, L. et al., 2010. Reinforcement learning and dynamic programming using function approximators. , p.260.

Dupont, B., De Jonghe, C., et al., 2014. Demand response with locational dynamic pricing to support the integration of renewables. *Energy Policy*, 67, pp.344–354.

Dupont, B., Dietrich, K., et al., 2014. Impact of residential demand response on power system operation: A Belgian case study. *Applied Energy*, 122, pp.1–10.

Herter, K., McAuliffe, P. & Rosenfeld, A., 2007. An exploratory analysis of California residential

customer response to critical peak pricing of electricity. *Energy*, 32(1), pp.25–34.
IEA, 2013. *Transition to Sustainable Buildings*, Available at: https://www.iea.org/publications/freepublications/publication/Building2013_free.pdf.
Kalyanakrishnan, S., Stone, P. & Liu, Y., 2008. Batch Reinforcement Learning in a Complex Domain. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5001 LNAI, pp.171–183.
McNeil, M. a. & Letschert, V.E., 2008. Future air conditioning energy consumption in developing countries and what can be done about it: the potential of effi ciency in the residential sector. Available at: http://www.eceee.org/library/conference_proceedings/eceee_Summer_Studies/2007/Panel_6/6.306/paper.
Mozer, M.C., 1998. The Neural Network House: An Environment that Adapts to its Inhabitants. *American Association for Artificial Intelligence Spring Symposium on Intelligent Environments*, (December), pp.110–114.
Rault, A., 1978. Model Predictive Heuristic Control : Applications to Industrial Processes *. , 14, pp.413–428.
Robinson, D., 2011. *Computer modelling for sustainable urban design*, London: Earthscan.
Ruelens, F. et al., 2014. Demand response of a heterogeneous cluster of electric water heaters using batch reinforcement learning. *Proceedings - 2014 Power Systems Computation Conference, PSCC 2014*, (2).
Shaikh, P.H. et al., 2014. A review on optimized control systems for building energy and comfort management of smart sustainable buildings. *Renewable and Sustainable Energy Reviews*, 34, pp.409–429.
Siano, P., 2014. Demand response and smart grids - A survey. *Renewable and Sustainable Energy Reviews*, 30, pp.461–478. Available at: http://dx.doi.org/10.1016/j.rser.2013.10.022.
De Somer, O. et al., 2017. Using Reinforcement Learning for Demand Response of Domestic Hot Water Buffers: a Real-Life Demonstration. , pp.1–6.
Summit Blue Consulting, L., 2007. Evaluation of the 2006 Energy-Smart Pricing Plan. Final Report. , pp.1–15. Available at: http://assets.fiercemarkets.net/public/smartgridnews/2006-espp-evaluation.pdf.
Sun, Y., Somani, A. & Carroll, T.E., 2015. Learning Based Bidding Strategy for HVAC Systems in Double Auction Retail Energy Markets Yannan. *Proceedings of the American Control Conference*, 2015–July, pp.2912–2917.
Vázquez-Canteli, J.R., Kämpf, J. H., Nagy, Z.G., 2017. Balancing comfort and energy consumption of a heat pump using batch reinforcement learning with fitted Q-iteration. *Energy Procedia*, 122, pp.415–420.
Vázquez-Canteli, J.R., Nagy, Z.G., 2018. Reinforcement learning for demand response: A review of algorithms and modeling techniques. , Under revi.
Vázquez-Canteli, J.R., Ulyanin, S., Kämpf, J. H., Nagy, Z.G., 2018. CityLearn: Fusing Deep Reinforcement Learning with Urban Energy Modelling for Adaptive Building Energy Control in Smart Cities. , Under revi.
Vázquez-Canteli, J.R. & Kämpf, J., 2016. Massive 3D models and physical data for building simulation at the urban scale : a focus on Geneva and climate change scenarios. *WIT Transactions on Ecology and the Environment*, 204(November).
Walter, E. & Kämpf, J.H., 2015. A verification of CitySim results using the BESTEST and monitored consumption values. *Proceedings of the 2nd Building Simulation Applications conference*, pp.215–222.
Watkins, C.J.C.H. & Dayan, P., 1992. Technical Note: Q-Learning. *Machine Learning*, 8(3), pp.279–292.
Yang, L., Nagy, Z., Goffin, P., Schlueter, A., 2015. Reinforcement learning for optimal control of low exergy buildings. *Applied Energy*, 156, pp.577–586.

NOMENCLATURE

S	states	Q	action-value
γ	discount factor	A	actions
V	state-value	π	policy
R	rewards	α	learning rate
ANN	Artificial Neural Network		
S’	States at the following time-step		
P	transition probability		
MDP	Markov Decision Process		
T	Boltzmann exploration constant		
RL	Reinforcement Learning		
BRL	Batch Reinforcement Learning		