Multi-agent reinforcement learning for adaptive demand response in smart cities

Multi-agent reinforcement learning for adaptive demand response in smart cities

Vazquez-Canteli, Jose[1]; Detjeen, Thomas[2]; Henze, Gregor[3]; Kämpf, Jérôme[4]; Nagy, Zoltan[1]

[1]The University of Texas at Austin, 301 E Dean Keeton St Stop C1700, 78712 Austin, TX, USA; [2]Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA; [3]University of Colorado Boulder, Boulder, CO 80309; USA; [4]Idiap Research Institute, Centre du Parc, Rue Marconi 19, PO Box 592, CH-1920, Switzerland

E-mail: nagy@utexas.edu

Abstract. Buildings account for over 70% of the electricity use in the US. As cities grow, high peaks of electricity consumption are becoming more frequent, which leads to higher prices for electricity. Demand response is the coordination of electrical loads such that they react to price signals and coordinate with each other to shave the peaks of electricity consumption. We explore the use of multi-agent deep deterministic policy gradient (DDPG), an adaptive and model-free reinforcement learning control algorithm, for coordination of several buildings in a demand response scenario. We conduct our experiment in a simulated environment with 10 buildings.

1. Introduction

Buildings account for over 70% of the electricity use in the US [1], and increasing the degree of electrification in urban areas can help decrease carbon emissions [2]. Renewable energy resources can help reduce our need for fossil fuels, and distributed generation has the potential to make buildings less dependent on the electrical grid. However, integration of renewable energy resources has certain challenges regarding grid stability and security of supply. Demand response can help buildings play an active role in the generation and storage of electricity by increasing demand flexibility. Demand response programs must also allow the coordination of multiple buildings such that the net energy peaks are not only shaved but shifted. Model-based control approaches, such as Model-Predictive Control (MPC), can achieve good performance at the cost of investing resources to develop the models. However, buildings are dynamic energy systems, in constant change due to diverse factors, e.g., refurbishment measures, installation of PV panels, changes in the energy supply, or variations in consumption patterns. Furthermore, wholesale electricity prices depend on a wide variety of factors, many of which change over time and are difficult to model and predict accurately. Also, it may not be feasible to model many systems due to their level of complexity.

Adaptive model-free control approaches, such as reinforcement learning, have the potential to overcome these challenges, as they can constantly learn from, and adapt to, their changing environment. These control approaches minimize any biases that may occur at the model-development stage and can directly learn from historical data and be better suited to control complex environments for which a model would be too expensive to develop. Reinforcement learning (RL) has been used in the past to

control building energy systems [3] and other urban energy systems. However, the use of multi-agent RL in urban energy systems for demand response purposes is still widely unexplored [4].

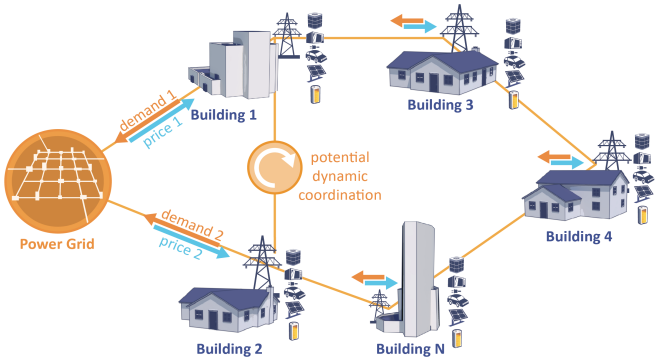

Figure 1: Dynamic coordination among buildings in demand response.

In this research, we explore the use of reinforcement learning (RL), an adaptive and potentially model-free control algorithm, for multi-agent coordination of several buildings in a demand response scenario. Specifically, we implement the multi-agent deep deterministic policy gradient (MADPG) [5], a state-of-the-art RL algorithm, which is superior to standard Q-learning RL, and has the advantages of being model-free and the ability to learn from historical data. Building on our previous work [6][7], we implement this controller in a simulated case study of 10 buildings with storage devices, and variable demand-dependent electricity prices. Outdoor temperatures, state of the charge, hour of the day, prices of electricity are used as states by the controllers.

2. Methodology

Reinforcement learning (RL) is formalized using a Markov Decision Process (MDP). MDPs contain: a set of states S, a set of actions A, a reward function $R: S \times A$ and transition probabilities between the states $P: S \times A \times S \in [0,1]$. The policy π represents a mapping between states and actions, $\pi: S \rightarrow A$, and the value function $V^{\pi}(s)$ is the expected return for the agent when starting in the state s and following the policy π, i.e.,

$$V^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s\prime} P^{a}_{s\,s\prime}[\, R^{a}_{s\,s\prime} + \gamma V^{\pi}(s')] \tag{1}$$

where $R^{a}_{s\,s\prime}$, which can be denoted as $r(s,a)$, is the reward obtained after taking an action $a = \pi(s_k)$, and transitioning from the current state s to the next state s', and $\gamma \in [0,1]$ is a discount factor for future rewards. An agent that uses $\gamma = 1$ will consider future rewards as important as current rewards, whereas for $\gamma = 0$, greater values are assigned to states that lead to high immediate rewards. RL is particularly useful when the model dynamics (P and R) are unknown and must be estimated through interaction of the agent with the environment (as depicted in Figure 2). There are two approaches that can be used to determine the values V^{π} of every state:

- A model-based approach, in which the rewards and transition probabilities of the model are first learned, and then used to find the values by solving iteratively the system of equations represented by Equation (1).
- A model-free approach, the agent learns the values associated to every (s,a) pair without explicitly calculating the transition probabilities or the expected rewards [8][9].

Q-learning is the most widely used model-free RL technique due to its simplicity [10]. For tasks with small finite state sets, all the different transitions can be represented with a table that stores the

state–action values, or Q-values. Each entry in the table represents a state–action tuple (s, a), and the Q-values are updated as

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r(s,a) + \gamma\, max_a\, Q(s',a) - Q(s,a)] \quad (2)$$

where s' is the next state, and $\alpha \in (0,1)$ is the learning rate, which explicitly defines to what degree new knowledge overrides old knowledge. For $\alpha = 0$, no learning happens, while for $\alpha = 1$, prior Q-values are completely overridden by the new Q-values in every iteration.

2.1. Deep deterministic policy gradient

In order to control complex environments that have many continuous states and actions, tabular Q-learning is not useful, as it faces a problem that becomes non-tractable due to the curse of dimensionality. Other RL algorithms, such as the deep deterministic policy gradient (DDPG), are better suited for these types of problems. DDPG is an actor-critic method, which uses two deep artificial neural networks (DNN) to generalize across the state-action space. The actor network maps the current states to the actions that it estimates to be optimal. Then, the critic network evaluates those actions by mapping them, together with the states under which they were taken, to the Q-values, as Figure 2 illustrates. The Q-values represent the expected cumulated sum of the discounted rewards after taking an action under a certain state and following the greedy policy thereafter.

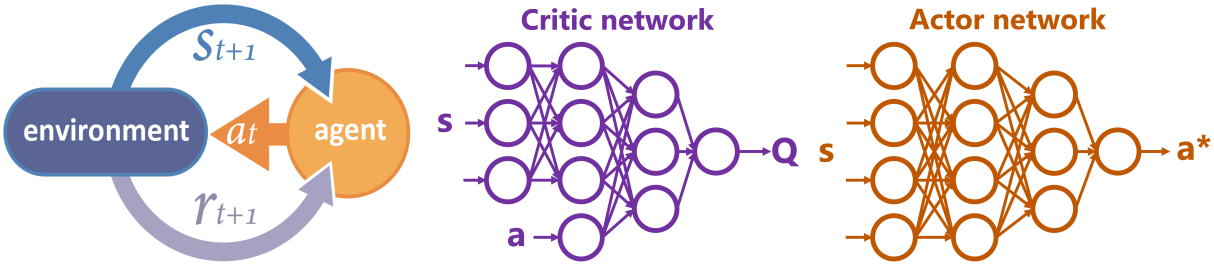

Figure 2: Agent-environment interaction, and actor and critic networks in reinforcement learning.

The critic network is updated by fitting the Q-values calculated from Eq. 2 with $\alpha = 1$, and minimizing the loss between the Q-values and their predictions (Eq. 3). While the actor network is updated using the gradient of the Q-function with respect to the parameters of the actor network as Eq. 4 represents.

$$L(\theta^Q) = \frac{1}{N}\sum (Q(s_t, a_t|\theta^Q) - y_t)^2 \quad (3)$$

$$\nabla_{\theta\mu} J = \frac{1}{N}\sum \nabla_a Q(s, a|\theta^Q)|s=s_i, a= \mu(s_{t+1})|\theta^Q = \frac{1}{N}\sum \nabla_a Q(s, a|\theta^Q)_{|s=si,\, a=\mu(si)}\, \nabla_{\theta\mu}\mu(s|\theta^\mu)|s_i \quad (4)$$

2.2. Multi-agent coordination

Every building has its own chilled water tank and heat pump, as depicted in Figure 3. The heat pumps provide cooling energy to the water tanks, which satisfy the cooling loads of the buildings. If the tank cannot meet the cooling load of the building, the heat pump provides the extra cooling energy to ensure that the temperature set point of the building is always met. The temperature set points of the buildings are assumed to be constant. The heat pumps consume electricity from the power grid, and the price of electricity increases linearly with the total electricity demand of all the buildings. Therefore, the objective of the controllers is to reduce their cost of electricity by avoiding consuming too much electricity simultaneously and consuming less energy.

Every building has its own RL-DDPG controller, and there are as many agents or controller as number of buildings (10 of them in this case). Two different RL controllers were simulated: a DDPG controller, and its multi-agent variant MADPG. In DDPG, every agent only knows its own states and actions, while in MADPG, every agent also knows the states of all the other agents. As states we used the hour of the day, the outdoor temperature, and the state of charge in every chilled water tank.

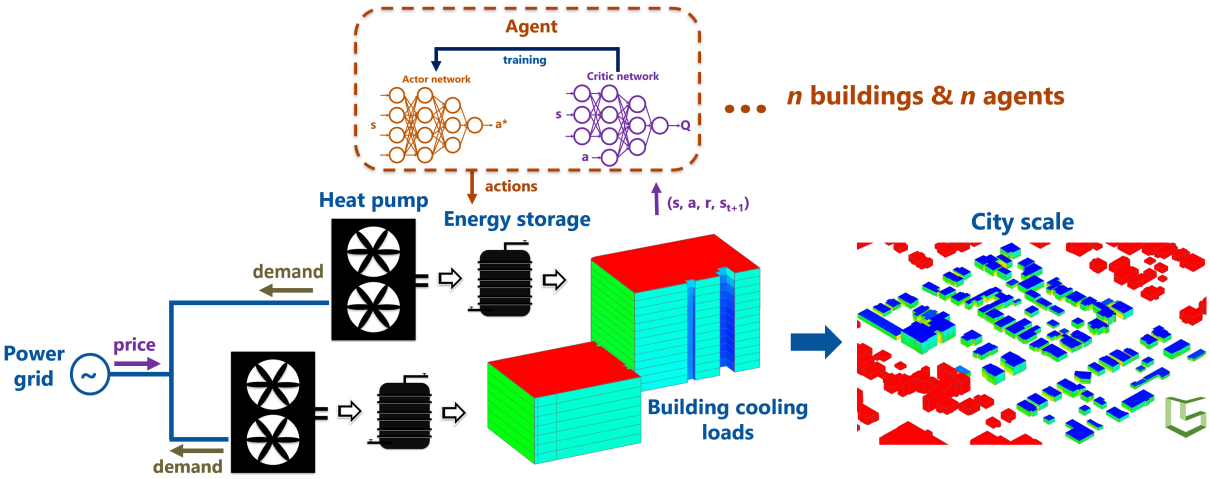

Figure 3: Implementation of MADPG in a simulated demand response scenario.

Space cooling is generally done by consuming electricity and in hot climates such as Austin, TX, cooling causes the highest annual peaks in electricity use and prices. Therefore, in this study we have focused on controlling only the cooling devices of 10 buildings in Austin, TX, over the summer period (June-August), and with no prior knowledge of the system for the controller. Regarding the computational load, the simulation with the reinforcement learning controller lasted about 40 minutes (CPU: i7-6700 K 4.0 GHz, RAM: 64.0 GB).

The actions of the controller are the energy they store or release in every hourly interval. The energy demand of the buildings is simulated with CitySim, a building energy simulator for urban scale analysis, while the electricity prices increase linearly with the total electricity demand. Buildings have diverse characteristics, energy consumption patterns, and diverse generation or storage devices such as heat pumps, and water tanks. Furthermore, we have developed our simulation environment as an OpenAI Gym Environment [11], which increases standardization, reproducibility of the results, and makes RL algorithms easier to implement. Many environments in which RL algorithms are tested for benchmarking purposes use this format.

3. Results

We used the total cost of cooling as the performance metric to evaluate the controllers. As Figure 4 illustrates, we compare the DDPG and MADPG controllers with two rule-based controllers (RBC). One is a standard rule-based controller that cools the water in the tank once it reaches a maximum temperature threshold, while the other RBC has been manually tuned to minimize the total cost of electricity. Both the DDPG and the MADPG outperformed the standard RBC. This improvement in performance happened without the need for modelling any of the systems being controlled and maintaining the adaptive potential of RL. The manually optimized RBC performed better than both the DDPG and the MADPG controllers, although it was relatively easy to tune because all the buildings had very similar energy systems. As the complexity of these environment increases, with the addition of PV panels, batteries, use of appliances, and energy storage in the thermal mass of the buildings, we expect that the DDPG and MADPG would outperform any RBC [7]. More complex systems would require a more

detailed understanding of their behavior, and possibly modelling them, which could be avoided by using RL.

As Figure 4 shows, in this environment the MADPG controller did not seem to have any advantage over the DDPG controller. Therefore, sharing information among the agents did not led to any improvements in performance. This would mean that no attempt for coordination needs to be made, while similar savings can be realized. It remains to be researched further whether this is also the case in a more complex environment with more diverse energy systems to control, and in which buildings may have very different optimal policies.

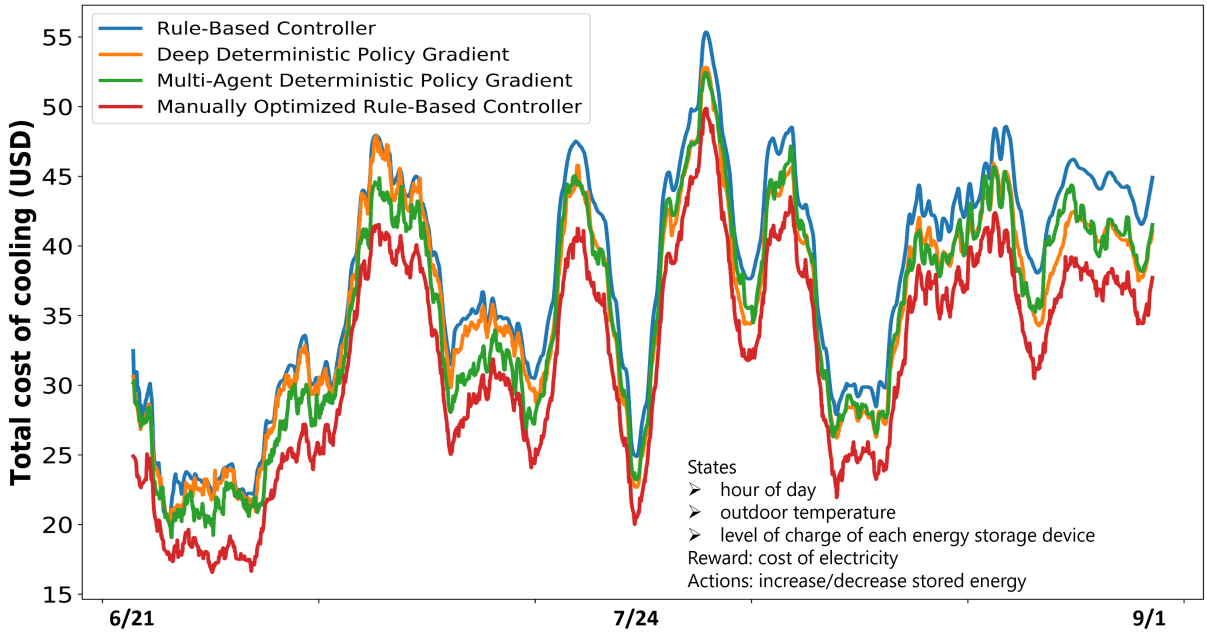

Figure 4: Total cost of cooling for 10 buildings under the weather conditions of Austin, TX.

The 10 buildings were simulated for the Summer period (June-August), and Figure 4 illustrates the results of the last month of this period, after the algorithm had spent 2 months learning by interacting with its environment using a tradeoff between taking exploratory actions and exploiting the actions that it considers to be optimal. We observed that after learning for a month, the RL controllers were already able to improve their performance significantly over the RBC.

4. Conclusion

Demand response enables buildings and other urban energy systems to respond to price signals when they consume electricity from the power grid. As a result, urban energy systems can coordinate with each other to consume energy more efficiently and avoiding consuming it simultaneously. Reinforcement learning control methods, such as DDPG and its multi-agent variant MADPG, can effectively coordinate urban energy systems in an adaptive way and without the need for developing any models. Its model-free nature and adaptive capabilities makes RL-based control methods particularly advantageous to control systems that are in constant change, such as buildings, or for which developing a model is too costly or difficult.

In this research, we have shown how ten RL controllers were able to learn from their respective buildings and their surrounding neighbors and achieve successful coordination to reduce their cost of electricity for cooling. Our future research will focus on increasing the complexity of these urban energy systems, by adding PV panels, batteries and energy storage in the thermal mass of the buildings, in order to prove how RL is most advantageous in more complex environments.

Finally, we are currently working on publicly releasing the OpenAI Gym environment of these urban energy systems with the aim of increasing standardization, reproducibility of the results, and encouraging other researchers to design their RL controllers for demand response applications.

References

[1] Society AP. Buildings. Energy Futur. Think Effic., 2008, p. 52–85.
[2] Leibowicz BD, Lanham CM, Brozynski MT, Vázquez-Canteli JR, Castillo N, Nagy Z. Optimal decarbonization pathways for urban residential building energy services. Appl Energy 2018;230:1311–25. doi:10.1016/j.apenergy.2018.09.046.
[3] Liu S, Henze GP. Evaluation of Reinforcement Learning for Optimal Control of Building Active and Passive Thermal Storage Inventory. J Sol Energy Eng 2007;129:215. doi:10.1115/1.2710491.
[4] Vázquez-Canteli JR, Nagy Z. Reinforcement learning for demand response : A review of algorithms and modeling techniques. Appl Energy 2019;235:1072–89. doi:10.1016/j.apenergy.2018.11.002.
[5] Lowe R, Tamar A, Jan LG, Harb J, Abbeel P, Mordatch I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments **arXiv:1706.02275v3 [cs.LG]**
[6] Vázquez-Canteli J, Kämpf J, Nagy Z. Balancing comfort and energy consumption of a heat pump using batch reinforcement learning with fitted Q-iteration. Energy Procedia 2017;122:415–20. doi:10.1016/j.egypro.2017.07.429.
[7] Vázquez-canteli JR, Ulyanin S, Kämpf J, Nagy Z. Fusing TensorFlow with building energy simulation for intelligent energy management in smart cities. Sustain Cities Soc 2019;45:243–57. doi:10.1016/j.scs.2018.11.021.
[8] Sutton R, Barto A. Reinforcement Learning: An Introduction. MIT Press Cambridge, Massachusetts 1998.
[9] Z. Nagy, J. Y. Park, and J. Vazquez-Canteli, "Reinforcement learning for intelligent environments: A Tutorial," in *Handbook of Sustainable and Resilient Infrastructure*, 1st ed., P. Gardoni, Ed. Routledge, 2018.
[10] Watkins CJCH, Dayan P. Technical Note: Q-Learning. Mach Learn 1992;8:279–92. doi:10.1023/A:1022676722315.
[11] Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, et al. OpenAI Gym: **arXiv:1606.01540v1 [cs.LG]**