Building HVAC control using reinforcement learning

Tanmay Ambadkar
Department of Computer Science and Engineering
Pennsylvania State University
University Park, PA, 16803
ambadkar@psu.edu

Rosina Adhikari
Department of Architectural Engineering
Pennsylvania State University
University Park, PA, 16803
rosina.adh@psu.edu

Abstract

Buildings have multiple HVAC systems designed to control their internal temperature. They run on fossil fuels and electricity, consume a lot of energy, and thus are expensive to operate. Reports suggest that buildings consume 35% of the total energy in the United States, and 60% is consumed for heating and cooling. Thus, it is important to operate them efficiently to reduce the operating costs. Currently used techniques are rule-based and not very efficient. In this paper, we propose a reinforcement learning-based solution to optimize the energy consumption of a building. We propose a shared-critic PPO algorithm that is used for three tasks that finally optimize the energy consumption of the building. We compare with several baselines to show the efficacy of our solution

As we progress towards a carbon-neutral future, we must reduce the energy consumed by our buildings since buildings account for more than 30% of global energy consumption. Heating, Ventilation, and Air- Conditioning (HVAC) systems are the most significant contributor to the total energy consumed in a building. Reducing the energy consumed by HVAC systems is not straightforward since it is directly associated with the comfort and well-being of human occupants inside the building. To improve energy efficiency in HVAC systems without compromising human comfort, the HVAC control strategies should be designed to adapt automatically to the dynamic conditions in a built environment (like outdoor temperature, occupancy status, energy price, etc.). However, traditional control methods are usually based on fixed rules or control algorithms, leading to suboptimal comfort and energy performance. Feedback-based controllers like Proportional-Integral-Derivatives (PID) controllers have limitations for non-linear, often fluctuating control applications. Other alternatives, such as Model Predictive Control (MPC), can adapt to dynamic needs. Still, the stochastic nature of occupancy and the scale of the building's thermodynamic model adds tremendous complexity to the formulation and computation of MPC-based control strategies. Therefore, Reinforcement learning (RL), the model-free control strategy, has recently gained attention as a promising solution for complex control problems, including HVAC systems control.

Our contributions can be summarised as follows:

- Break down the reinforcement learning problem into multiple tasks
- Propose a multi-task PPO algorithm to solve multiple tasks

1 Related works

The literature mainly applies the combination of deep neural networks and RL (DRL) to HVAC systems control applications. Wang et al. developed a novel model-free reinforcement learning algorithm-based feedback control for HVAC systems with proven efficacy [1]. Zhang et al. proposed a practical control framework - BEM-DRL [2] that is based on deep reinforcement learning combined with building energy models. Gao et al. [3] proposed a deep reinforcement learning-based framework

Preprint. Under review.

for energy optimization and thermal comfort control in intelligent buildings. Xu et al. [4] presented a novel transfer learning-based approach to improve the scalability of DRL-based controls. Advancing further, Yu et al. [5] developed a multi-agent deep reinforcement learning with an attention mechanism to minimize the energy cost of an HVAC system in a multi-zone commercial building considering random zone occupancy, thermal comfort, and indoor air quality comfort. Du et al. [6] applied a novel model-free deep reinforcement learning (RL) method, known as the deep deterministic policy gradient (DDPG), to generate an optimal control strategy for a multi-zone residential HVAC system. These are some notable references for this project.

2 Formulating the MDP

The Markov Decision Process (MDP) consists of a tuple $\{S, A, R, P, \gamma\}$. An agent looks at state s_t and performs an action $a \in A$, upon which it reaches a new state s_{t+1} with probability $P_{s_t,a,s_{t+1}}$ and gets a reward $r_{s_{t+1},a}$. Building HVAC data is usually in the form of a time series. Thus, it makes it difficult to formulate an MDP. This is because it has no reward signal. Using an imitation learning-based framework would not help optimize the energy consumption because the time series data is for regular unoptimized HVAC operations. Thus, we need a simulation that can take an action and produce a new state with stochasticity, considering the exogenous variables like the weather and occupancy of the building.

Raboso et al. [7] have provided a framework to simulate the behavior of multiple buildings in a gymnasium environment. It uses EnergyPlus [8] as the host simulator and Building Controls Virtual Test Bed [9] to connect Python to the EnergyPlus API. This significantly reduces the task of formulating the MDP ourselves.

Sinergym [7] uses both external (weather, humidity) and internal factors (temperature) as the state and the heating and cooling setpoints as the action to control the HVAC. The heating and cooling setpoints control the operating temperature of the HVAC. This affects energy consumption. If the external temperature is high and the building must be cooled, then much energy is consumed. The ideal setting would be when the external temperature is comfortable for the human body, and the HVAC need not be operated. Since this is not the case, the HVAC is turned on most of the time, and energy is consumed.

Simply reducing energy consumption is not the goal. We need to consider the comfort of the occupants in the building. Thus, the reward signal consists of the energy consumed and the comfort of the occupants. Both rewards are negative, meaning perfect behavior is when the reward is 0. The reward is formulated as Equation 1.

$$R_t = -w\lambda_E E_t - (1-w)\lambda_T(|T_t - T_{heat}| + |T_t - T_{cool}|) \tag{1}$$

It is a weighted reward signal. The environment gives more weight to the comfort parameter than the energy consumed, thus if we use an algorithm that takes this scalar reward into account, it does not perform very well.

3 Multi Task reinforcement learning

There are multiple reinforcement learning problems that involve solving multiple tasks at once. Most of these involve solving robotic control tasks. They learn compound policies and then share the knowledge to either learn a new policy for a new task or a global policy for all tasks. [10, 11] proposed sharing learned knowledge for tasks that have the same action space. [12] proposes to formulate state space for similar tasks such that they have the same structure, and they learn independently and share their knowledge via a shared critic. Our proposed multi-task algorithm heavily borrows from their work.

4 Multi Task Actor Critic

In [12], the authors propose a framework with multiple actor critics, each for a single task. These actor critics have the same state space, which is achieved by modifying the state space to match all

tasks. In addition to having task-specific actor critics, they have a shared critic that evaluates the performance of the learned task with other tasks. This is used for sharing the learned knowledge from other tasks. An illustration of the framework can be seen in Figure 1.

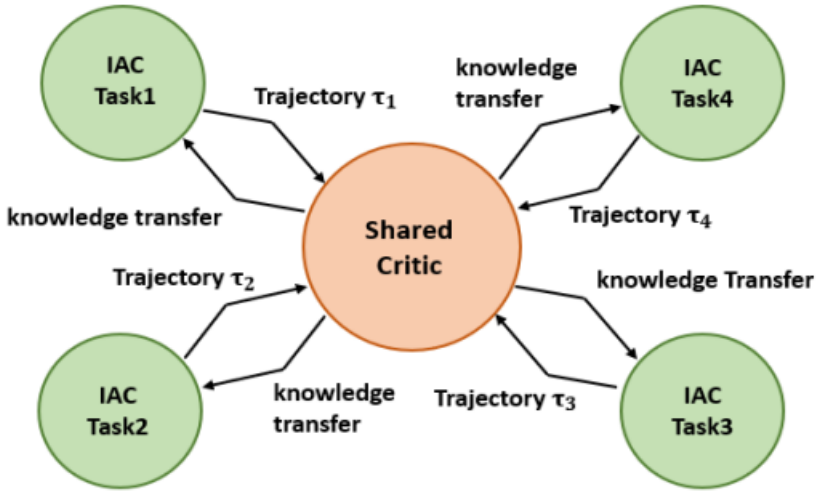

Figure 1: An illustration of the MultiTaskAC [12]

We see in Figure 1 that there are individual actor critics learning through their own trajectory. This trajectory is shared with the shared critic for knowledge transfer.

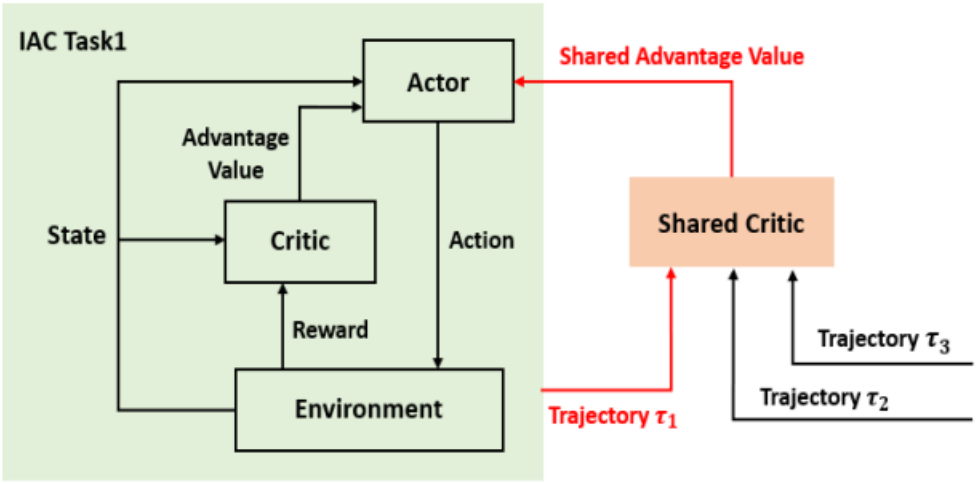

Figure 2: Capturing the IAC and shared critic

Figure 2 shows how the advantage value from the individual and shared critics is used to update the actor to modify its policy. Thus, there are two advantages used to update the actor, and the shared critic uses n trajectories to update its weights.

3 shows the algorithm used to update the MultiTaskAC. We see that there are three gradients computed: the individual actor and critic gradient and the shared critic gradient. The shared critic gradient is updated after updating the individual actor-critic gradient. [12] use the A2C algorithm for their work. We propose using the proximal policy optimization [13] instead of the A2C.

Algorithm 1 Multi-task actor-critic with a shared critic

Input: State s: Reward r

Parameter: Task number $i = 1, 2...M$; Number of Episode E; Maximum steps of task per episode T; Transfer weight α;

Output: Critic $\eta_{i,E}$ actor $\theta_{i,E}$;

Randomly initialise actor π_{θ_i} and critic network $V_{\eta_{ni}}$ for task i, $i = 1, 2...m$;
Initialize episode counter e, $e = 0$
while $e_i < E_i$ **do**
  for each task i **do**
    Set step counter $t = 0$
    while $t < T_i$ and not terminal state **do**
      Select action $a_{i,t} \sim \pi_{\theta_i}$.
      Execute $a_{i,t}$ and state $r_{i,t+1}$ and $s_{i,t+1}$
      Store tuple $(s_{i,t}, a_{i,t}, r_{i,t+1}, s_{i,t+1})$
      Update step counter: $t \leftarrow t + 1$
    end while
    for each sample **do**
      Compute advantage value using Eq.10
      Compute shared advantage value using Eq.8
    end for
    Compute critic gradient using Eq.12
    Compute actor gradient using Eq.11
    Compute shared critic gradient using Eq.6
  end for
  Update episode counter $e \leftarrow e + 1$
end while
return Critic and actor weights: $\theta_{i,E}, \eta_{i,E}$;

Figure 3: Algorithm used to update the MultiTaskAC

5 Proposed Solution

We propose using the MultiTaskAC [12] by replacing the A2C framework with PPO. We break down the HVAC control problem into three tasks. The first task is to optimize the energy. The reward for this task is $\lambda_E E_t$. The second task is to optimize the comfort of the occupants. The reward for this task is $\lambda_T(|T_t - T_{heat}| + |T_t - T_{cool}|)$. The third task is the original task of optimizing both. The intuition is that if two separate policies are learned to optimize energy and comfort, their knowledge can be transferred to an agent that learns a policy to optimize both.

The state space is the current state of the variables. We can use a Multi-layer perceptron network for the actor and critic. Since the inherent nature of this problem is that it depends on the historic data too, we can use that to modify the state space. We borrow the modeling philosophy from [14]. They do not take a sliding window to alter the time series forecasting task in their work. They take the values for a particular instance in time. For example, they take the values for Mondays at 12:00 noon for n such Mondays to forecast for the next $n + 1$ Monday at 12:00 noon. The reasoning is that the occupancy information and other factors are very similar across days of the week and the time. We use this philosophy along with a 1D convolutional network to create the actor-critic. Along with this, we test with 3 baselines. The first baseline is a simple MLP-based PPO. The next is a CNN-based PPO. To verify our implementation, the third baseline is the [15] PPO.

6 Results

This section is divided into 3 subsections. The first subsection shows the performance of all agents by using the reward function. The second subsection shows the performance of the agents using the comfort violation parameter. The third subsection shows the performance of the agents with the energy.

6.1 Rewards

The reward is the most important part of any reinforcement learning work. However, we do not focus on the reward because our goal is optimizing the energy. The reward from sinergym focuses on comfort more than energy.

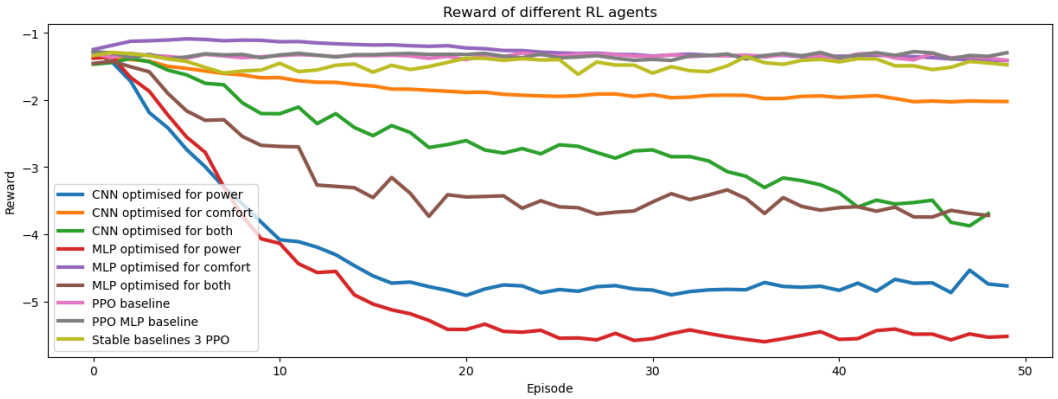

Figure 4: Reward over episodes for different agents

We can see in Figure 4 that the MLP optimized for power achieves the worst reward, followed by the CNN optimized for power. This is because they violate the comfort in order to achieve the lowest power consumption, thus they perform the worst in terms of reward. They are followed by the MLP and CNN optimized for both power and comfort.

6.2 Comfort violation

Comfort violation is the difference between the required temperature based on the occupants and the current temperature, expressed as a percentage. If the comfort violation is low, then the energy consumed is high.

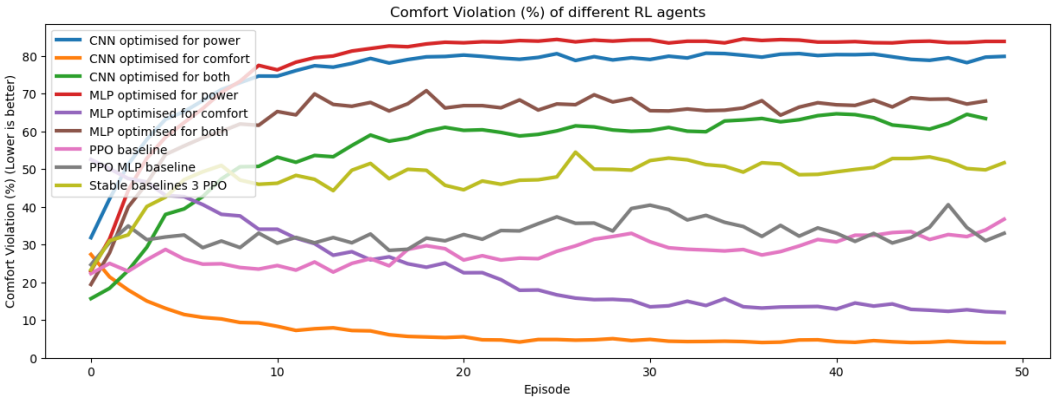

Figure 5: Comfort violation over episodes for different agents

In Figure 5, the MLP and CNN optimized for power have the highest comfort violation. The best results are for the MLP, and CNN optimized for comfort. The baselines lie in between, at 50% comfort violation.

6.3 Power consumption

The power consumption for every episode is the mean of the power consumption for that episode. If the comfort violation was high, the power consumption was low. This is because power was not utilized to turn the HVACs on.

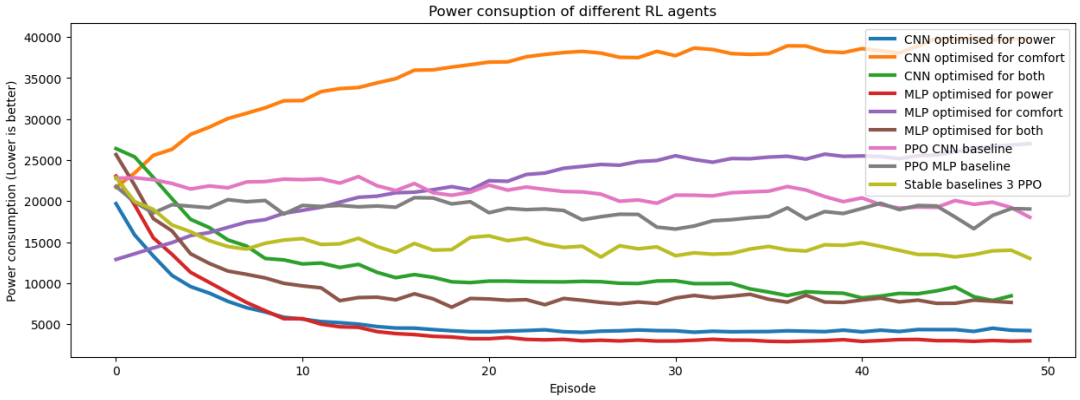

Figure 6: Power consumption over episodes for different agents

In Figure 6, the CNN and MLP optimized for comfort have the highest power consumption. The MLP and CNN optimized for both have similar power consumption over episodes and are far better optimized than other baselines.

7 Discussion of results

The results are a clear indicator that one task will dominate the other task based on the weight parameter in the reward. The baselines show that they prefer optimizing for comfort rather than power to achieve the lowest possible reward, which is not optimal because the power draw is very high. Our MultiTaskPPO shows that learning to optimize for both tasks individually results in the global policy for optimizing both to perform far better than the baselines, meaning that we are able to transfer learned knowledge to the global policy. Thus, this task requires learning two separate tasks before we can optimize using the original reward.

8 Conclusion

In this work, we have approached the problem of optimizing building HVAC operating costs by formulating it as a Markov Decision Process (MDP) and training a Multi-Task PPO policy that learns to optimize power and comfort individually and then transfers its learned knowledge to a global PPO policy using a shared critic. Several baselines are chosen to compare with to show the efficacy of the multi-task PPO policy.

References

[1] Yuan Wang, Kirubakaran Velswamy, and Biao Huang. A novel approach to feedback control with deep reinforcement learning∗∗this work is supported by in part by mitacs, alberta innovates and natural sciences engineering research council of canada. *IFAC-PapersOnLine*, 51(18):31–36, 2018. 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018.

[2] Zhiang Zhang, Adrian Chong, Yuqi Pan, Chenlu Zhang, and Khee Poh Lam. Whole building energy model for hvac optimal control: A practical framework based on deep reinforcement learning. *Energy and Buildings*, 199:472–490, 2019.

[3] Guanyu Gao, Jie Li, and Yonggang Wen. Deepcomfort: Energy-efficient thermal comfort control in buildings via reinforcement learning. *IEEE Internet of Things Journal*, 7(9):8472–8484, 2020.

[4] Shichao Xu, Yixuan Wang, Yanzhi Wang, Zheng O'Neill, and Qi Zhu. One for many: Transfer learning for building hvac control. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys '20, page 230–239, New York, NY, USA, 2020. Association for Computing Machinery.

[5] Liang Yu, Yi Sun, Zhanbo Xu, Chao Shen, Dong Yue, Tao Jiang, and Xiaohong Guan. Multi-agent deep reinforcement learning for hvac control in commercial buildings. *IEEE Transactions on Smart Grid*, PP:1–1, 07 2020.

[6] Yan Du, Helia Zandi, Olivera Kotevska, Kuldeep Kurte, Jeffery Munk, Kadir Amasyali, Evan Mckee, and Fangxing Li. Intelligent multi-zone residential hvac control strategy based on deep reinforcement learning. *Applied Energy*, 281:116117, 2021.

[7] Javier Jiménez-Raboso, Alejandro Campoy-Nieves, Antonio Manjavacas-Lucas, Juan Gómez-Romero, and Miguel Molina-Solana. Sinergym: A building simulation and control framework for training reinforcement learning agents. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, page 319–323, New York, NY, USA, 2021. Association for Computing Machinery.

[8] Energyplus™, version 00, 9 2017.

[9] Michael Wetter, Philip Haves, Brian Coffey, and USDOE. Building controls virtual test bed, 4 2008.

[10] Parijat Dewangan, S. Phani Teja, K. Madhava Krishna, Abhishek Sarkar, and Balaraman Ravindran. Digrad: Multi-task reinforcement learning with shared actions. *CoRR*, abs/1802.10463, 2018.

[11] Haitham Bou Ammar, Jose Luna, Eric Eaton, and Paul Ruvolo. Autonomous cross-domain knowledge transfer in lifelong policy gradient reinforcement learning. 01 2015.

[12] Gengzhi Zhang, Liang Feng, and Yaqing Hou. Multi-task actor-critic with knowledge transfer via a shared critic. In Vineeth N. Balasubramanian and Ivor Tsang, editors, *Proceedings of The 13th Asian Conference on Machine Learning*, volume 157 of *Proceedings of Machine Learning Research*, pages 580–593. PMLR, 17–19 Nov 2021.

[13] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[14] Nisha Menon, Shantanu Saboo, Tanmay Ambadkar, and Umesh Uppili. Discrete sequencing for demand forecasting: A novel data sampling technique for time series forecasting. In *2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, pages 61–67, 2022.

[15] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.