Multi-Objective Deep Q-Networks for Domestic Hot Water Systems Control

Mohamed-Harith Ibrahim[1,2], Stéphane Lecoeuche[1], Jacques Boonaert[1] and Mireille Batton-Hubert[2]
[1]*IMT Nord Europe, Institut Mines-Télécom, Univ. Lille, Centre for Digital Systems, F-59000 Lille, France*
[2]*Mines Saint-Etienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, Institut Henri Fayol, F-42023, Saint Etienne, France*

Keywords: Multi-Objective Reinforcement Learning, Deep Reinforcement Learning, Electric Water Heater.

Abstract: Real-world decision problems, such as Domestic Hot Water (DHW) production, require the consideration of multiple, possibly conflicting objectives. This work suggests an adaptation of Deep Q-Networks (DQN) to solve multi-objective sequential decision problems using scalarization functions. The adaptation was applied to train multiple agents to control DHW systems in order to find possible trade-offs between comfort and energy cost reduction. Results have shown the possibility of finding multiple policies to meet preferences of different users. Trained agents were tested to ensure hot water production with variable energy prices (peak and off-peak tariffs) for several consumption patterns and they can reduce energy cost from 10.24 % without real impact on users' comfort and up to 18 % with slight impact on comfort.

1 INTRODUCTION

Different methods and techniques are used to control DHW systems. Optimization based methods and Reinforcement Learning (RL) are the most studied approaches in literature to adapt operations of systems to real needs. Authors in (Kapsalis et al., 2018) present an optimization based method to schedule the operation of an Electric Water Heater (EWH) for a given hot water consumption pattern under dynamic pricing and takes into account cost and comfort of users. In (Shen et al., 2021), authors propose an MPC-based controller to minimize electricity cost while maintaining comfort under uncertain hot water demand and peak/off-peak rate periods. MPC is an optimization based method that consists of modelling the system to be controlled, predicting its future behaviors and disturbances and controlling by taking actions that satisfy constraints and optimize desired objectives. The major drawback of optimization based approach is in the necessity of having a precise dynamic model of the system. Problems can arise because of the non adaptive nature of the model which can lead to sub-optimal performances.

On the other hand, multiple studies use RL to control DHW systems (Heidari et al., 2022) (Amasyali et al., 2021) (Ruelens et al., 2016) (Patyn et al., 2018) (Kazmi et al., 2018). In (Amasyali et al., 2021), authors train different agents using DQN to minimize electricity cost of water heater without causing discomfort to users. Their results are compared to other control methods and their approach outperforms rule-based methods and MPC based controllers. In (Heidari et al., 2022), authors suggest to use Double DQN to balance comfort, energy use and hygiene in DHW systems. The agent learns stochastic occupants' behaviors in an offline training procedure integrating a stochastic hot water model to mimic the use of occupants. The balance between these objectives is based on the design of the reward function which returns a single reward value.

To the best of author's knowledge, the existing works about DHW production control do not consider the conflicting nature of the studied objectives. For many decision problems that require the consideration of an important number of objectives, the increasing of performances of one objective may decrease the performances of other objectives. In addition, preferences over objectives can be expressed in multiple ways and may be different depending on users that are affected by the decision process.

Multi-Objective Reinforcement Learning (MORL) extends RL to problems with two or more objectives. Multiple studies adapt existing single-objective methods to a multi-objective context. Authors in (Van Moffaert et al., 2013) propose a

Ibrahim, M., Lecoeuche, S., Boonaert, J. and Batton-Hubert, M.
Multi-Objective Deep Q-Networks for Domestic Hot Water Systems Control.
DOI: 10.5220/0011647400003393
In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023) - Volume 3*, pages 234-242
ISBN: 978-989-758-623-1; ISSN: 2184-433X

general framework to adapt Q-learning to a multi-objective problems using a scalarization function that expresses preferences over different objectives. This can be done in a case of a prior articulation of preferences over objectives.

Some real-world problems are complex and may require the use of value function approximation to scale up tabular methods. DQN was presented in (Mnih et al., 2015) for single-objective cases. In this paper, we focus on solving multi-objective sequential decision problems by learning a single policy in a known preferences scenario with value-based methods. This is done by adapting DQN to solve multi-objective problems using scalarization functions. This method is used to train a controller to take decisions about DHW production. Its objectives are to maximize comfort and to minimize energy cost.

The reminder of the paper is structured as follows. Section 2 gives a brief introduction to MORL. In Section 3, we present an adaptation of DQN to multi-objective problems. The control of DHW production and the experimental setup are presented in Sections 4 and 5. Finally, results for DHW production control are given in Section 6.

2 MULTI-OBJECTIVE REINFORCEMENT LEARNING

2.1 Definition

MORL can be viewed as the combination of multi-objective optimization and RL to solve multi-objective sequential decision problems. It is a branch of RL that involves multiple, possibly conflicting objectives. As illustrated in Fig. 1, at each time step t, the agent, in a certain state s_t, interacts with its environment via an action a_t that changes its state to s_{t+1} and provides a reward vector $\mathbf{r}_{t+1}$ containing a reward element for each objective. The reward function is a vector function that describes a vector of m rewards instead of a scalar.

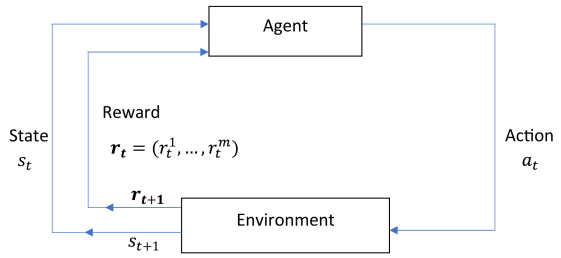


Figure 1: Agent-environment interaction in a multi-objective decision process.

Due to the vector reward function, state-value function V and action-value function Q under policy π are replaced by vector value functions $\mathbf{V}_\pi$ and $\mathbf{Q}_\pi$:

$$\mathbf{V}_\pi(s) = (V_\pi^{(1)}(s), \ldots, V_\pi^{(m)}(s)), \tag{1}$$

$$\mathbf{Q}_\pi(s,a) = (Q_\pi^{(1)}(s,a), \ldots, Q_\pi^{(m)}(s,a)), \tag{2}$$

where

$$V_\pi^{(i)}(s) = E_\pi[\sum_{n=0}^{T} \gamma^n r_{t+n+1}^{(i)} | s_t = s], \tag{3}$$

and

$$Q_\pi^{(i)}(s,a) = E_\pi[\sum_{n=0}^{T} \gamma^n r_{t+n+1}^{(i)} | s_t = s, a_t = a], \tag{4}$$

where T is the size of the sequence, m is the number of objectives, γ is the discount factor which is used to quantify the importance of short-term versus long-term rewards and $r^{(i)}$ is the reward for objective i.

The main goal for an agent in MORL problems is to optimize its expected cumulative rewards by learning a policy that best maps between states and actions.

2.2 Optimality in Multi-Objective Decision Problems

In multi-objective decision problems, no single policy exists that optimizes simultaneously all conflicting objectives. Instead, there exist a set of policies and one has to be chosen in the presence of trade-off between objectives. Therefore, to compare different policies and to define optimality in multi-objective problems, we use Pareto dominance relation as it was done in (Van Moffaert, 2016).

A policy π weakly Pareto dominates another policy π' when there does not exist an objective i where π' is better than π over all states:

$$\pi \succeq \pi' \iff \forall i, V_\pi^{(i)}(s) \geq V_{\pi'}^{(i)}(s). \tag{5}$$

Two policies are incomparable if some objectives have lower values for the first policy while others have higher values for the second policy and vice versa. Finally, a policy π is Pareto optimal if it either Pareto dominates or is incomparable to all other policies.

Multiple Pareto optimal policies could exist and the choice of a policy depends on the importance given to each objective. The set of Pareto optimal policies is called Pareto front.

2.3 Preferences over Objectives

Authors in (Liu et al., 2014) give a detailed overview of several MORL approaches. One way to express information about prioritizing objectives is to scalarize the multi-objective problem. Scalarizing means formulating a single-objective problem such that optimal policies to the single-objective problem are Pareto optimal policies to the multi-objective problem (Hwang and Masud, 2012). In addition, with different parameters quantifying the importance of each objective for the scalarization, different Pareto optimal policies are produced. Scalarizing in MORL means applying a scalarization function f and a weight vector $\mathbf{w}$ to the $\mathbf{Q}$-vector that contains Q-values of all objectives. This is done in the action selection stage in order to optimize the scalarized value expressed as follows:

$$SQ(s,a) = f(\mathbf{Q}(s,a),\mathbf{w}). \tag{6}$$

The scalarization function can be a linear scalarization function that computes a weighted sum of all Q-values. Other scalarization functions like Chebyshev scalarization (Van Moffaert et al., 2013) are also used in MORL. Besides, non-linear methods like Threshold Lexicographic Q-Learning (TLQ) (Vamplew et al., 2011) were proposed to learn a single policy in MORL. Nevertheless, some approaches may converge to a sub-optimal policy or even fail to converge in certain conditions as it was shown in (Issabekov and Vamplew, 2012) for TLQ. In fact, temporal-difference methods based on Bellman equation are incompatible with non-linear scalarization functions due to the non-additive nature of the scalarized returns (Roijers et al., 2013). Therefore, in what follows, we consider f as a linear scalarization function and $\mathbf{w}$ a weight vector such as:

$$f(\mathbf{Q}(s,a),\mathbf{w}) = \sum_{i=1}^{m} w_i Q^{(i)}(s,a), \tag{7}$$

where $\forall i, 0 \leq w_i \leq 1$ and $\sum_{i=1}^{m} w_i = 1$.

2.4 Action Selection in MORL

In value based RL methods, the optimal policy is derived from estimated Q-values by selecting actions with the highest expected cumulative rewards: we choose greedy actions.

$$a = \arg\max_{a'} Q(s,a'). \tag{8}$$

In MORL, the Pareto optimal policy is derived from estimated Q-vectors by selecting actions with the highest scalarized expected cumulative rewards: we choose scalarized greedy actions.

$$a = \arg\max_{a'} SQ(s,a'). \tag{9}$$

In order to balance exploration and exploitation, we use ε-greedy action selection. ε refers to the probability of choosing to explore by selecting random actions while $1-\varepsilon$ is the probability of exploiting by taking advantage of prior knowledge and selecting greedy actions.

3 MULTI-OBJECTIVE DEEP Q-NETWORKS

It is important to recall that DQN is about training a Neural Network (NN) with parameters θ to approximate the action-value function of the optimal policy π^*.

$$Q_\theta(s,a) \approx Q_{\pi^*}(s,a) = E_{\pi^*}[\sum_{n=0}^{T} \gamma^n r_{t+n+1} | s_t = s, a_t = a]. \tag{10}$$

The NN takes a state as an input and outputs the value of each possible action from that state. The method is characterized by:

- The use of a replay memory to store experiences.
- The use of two networks: a policy network Q_θ and a target network $Q_{\theta'}$. The policy network determines the action to take and is updated frequently by training on random batches from the replay memory. The target network is an old version of the policy network and is updated copying its weights from the policy network at regular intervals. It is used to compute targets $Q(s,a)$ noted y as follows:

$$y = r + \gamma max_{a'} Q_{\theta'}(s',a'). \tag{11}$$

The target is the estimated value of a state-action pair (s,a) under the optimal policy. It is the sum of the immediate reward r received after taking action a in state s and the estimated discounted maximum value from next state s'.

In this section we adapt DQN to multi-objective sequential decision problems. We suggest to train an NN Q_θ with parameters θ to approximate the action-value vector function $\mathbf{Q}$ of an optimal policy. The NN takes a state as an input and it outputs the value of each possible action for each objective from that state. The argument of separating Q-values for each objective instead of learning one scalarized Q-value is that values of individual objectives may be easier to learn than the scalarized one, particularly when function approximation is employed as mentioned in (Tesauro et al., 2007).

Similarly to DQN, we use a target network $Q_{\theta'}$ of parameters θ' to compute the targets. However, multiple changes are made to DQN:

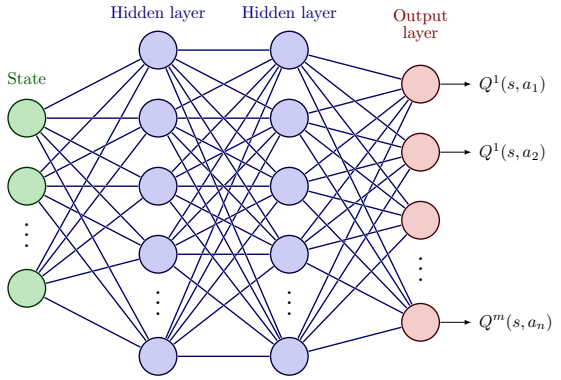


Figure 2: Architecture of a neural network to estimate Q-values for m objectives.

- The output layer of the trained NN outputs the value of each possible action for each objective. The size of the output layer becomes $n \times m$ where n is the number of possible actions at each time step (see Fig. 2).
- Action selection: the scalarization function f and the weight vector $\mathbf{w}$ are involved in the action selection process to express the importance of each objective. The greedy action becomes the action that guarantees the highest scalarized action-value as explained in Subsection 2.4.
- Replay memory: for each experience e, we store a reward vector $\mathbf{r}$ that contains a reward value for each objective. These experiences are used to train the value network.

$$e = (s, a, \mathbf{r}, s'). \tag{12}$$

- Target value computation is done for each objective i. The target is the sum of the immediate reward of the i'th objective and the i'th component of the $\mathbf{Q}$ vector of the scalarized greedy action a' from next state s' using the target network:

$$y^{(i)} = r^{(i)} + \gamma Q^{(i)}_{\theta'}(s', a'). \tag{13}$$

- The value network is trained to minimize the mean squared temporal difference error for each objective:

$$L(\theta) = \sum_{i=1}^{m} E[(y^{(i)} - Q^{(i)}_{\theta}(s,a))^2]. \tag{14}$$

In whats follows, we note $\mathbf{y}$ as the vector of target values for each objective.

Multi-Objective DQN (MO-DQN) is a single policy method that requires prior knowledge of preferences over different objectives. The method is summarized in Algorithm 1.

Algorithm 1: Multi-Objective DQN.

Initialize replay memory $\mathcal{D}$ to capacity N
Choose number of episodes M and episode length T
Choose learning rate α, discount factor γ and batch size B
Choose scalarization function f and weight vector $\mathbf{w}$
Initialize value network Q_θ with random weights θ
Copy the value network to create the target network $Q_{\theta'}$
for episode=1, M **do**
    Get an initial state
    for $t = 1, T$ **do**
        With probability ε select random action a_t
        Otherwise $a_t = \arg\max_a f(Q_\theta(s_t, a), \mathbf{w})$
        Execute action a_t and get rewards $\mathbf{r}_{t+1}$ and next state s_{t+1}
        Store experience $(s_t, a_t, \mathbf{r}_{t+1}, s_{t+1})$ in $\mathcal{D}$
        Move to next state s_{t+1}
        Sample a random batch $\mathcal{D}_B$ of size B of experiences from $\mathcal{D}$ every T_{train} steps
        for each experience $(s, a, \mathbf{r}, s') \in \mathcal{D}_B$ **do**
            Calculate $\mathbf{Q}_{\theta'}(s', a')$ the $\mathbf{Q}$ vector of the scalarized greedy action a' from the next state s' using the target network $Q_{\theta'}$
            Calculate expected state-action pair (s, a) values

$$\mathbf{y} = \mathbf{r} + \gamma \mathbf{Q}_{\theta'}(s', a')$$

        end for
        Train value network Q_θ on $\mathcal{D}_B$ to minimize the loss function expressed in Equation (14)
    end for
    Update ε for exploration probability
    Update target network's weights θ' with the weights of the value network every K step
end for

4 DOMESTIC HOT WATER PRODUCTION PLANNING AND CONTROL

In this section, we study the control of an EWH. The goal is to train a controller using MO-DQN to take decisions about DHW production considering users' comfort and energy cost. Preferences over these two objectives may be different from householder to another. In addition, increasing comfort may increase energy cost. Thus, decision-making in this case is about finding a trade-off between conflicting objec-

tives based on preferences over objectives.

The controller has to take a decision about water heating for the next time step based on information at current time step. In addition, the decision process considers importance given to each objective using a scalarization function f and a weight vector $\mathbf{w}$.

4.1 State Representation

The state vector s_t is a representation of the environment at time t. It contains time-related components, such as hour of the day h and day of the week d, temperature measurement T, DHW consumption V_{DHW} and electricity tariff λ:

$$s_t = (h(t), d(t), T(t), V_{DHW}(t), \lambda(t)), \quad (15)$$

$s_t \in \mathcal{S}$ where $\mathcal{S}$ is the state space.
The time-related information helps the agent to associate repeated behaviors to time without requiring prediction of DHW consumption. In fact, as shown in (Heidari et al., 2021), hot water use behaviors are highly correlated with the same time of the day and the behaviors during the weekdays can be similar and different from the weekends.

For energy cost, we use french electricity tariffs in early 2022 with two periods : off-peak time and peak time. The price of 1 kWh is in euro and is 25.1% more expensive in peak time:

$$\lambda = \begin{cases} 0.147 & \text{from 12 am to 8 am,} \\ 0.184 & \text{from 9 am to 11 pm.} \end{cases} \quad (16)$$

4.2 Control Actions

At each time step t, the agent takes an action $a_t \in \mathcal{A}$ where $\mathcal{A} = \{0, 20, 40, 60\}$ is the action space. The action is taken each hour ($\Delta t = 60$ minutes) based on the current state and it represents the duration, in minutes, of production at time step t. We assume that the EWH has a rated power P_{elec} of 2.2 kW.

4.3 Reward Shaping

To minimize energy cost, we design a cost reward where the agent is penalized each time it decides to produce DHW. The reward takes into consideration the duration of production and electricity tariff. This would encourage the agent to shift DHW to periods where energy is less expensive and to reduce its energy consumption by reducing the duration of DHW production. The received reward after a decision a_t at state s_t for energy cost is:

$$r_{t+1}^{cost} = -\frac{a_t}{\Delta t} \times P_{elec} \times \lambda(t). \quad (17)$$

In order to avoid discomfort situations, we design a comfort reward where the agent is penalized each time the temperature of DHW is lower than a minimum threshold accepted by the user called T_{pref}. This would motivate the agent to stay in a state where water temperature is acceptable for the user. The received reward after a decision a_t at state s_t for comfort is:

$$r_{t+1}^{comfort} = \begin{cases} 0 & \text{if } T(t+1) \geq T_{pref}, \\ -10 & \text{otherwise.} \end{cases} \quad (18)$$

Both rewards are normalized to have a common scale. The importance of each objective is expressed using the scalarization function f in the action selection process. Thus, the reward function $\mathcal{R}$ is defined as follow:

$$\mathcal{R}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^2 \quad (19)$$
$$(s, a, s') \longmapsto (r^{comfort}, r^{cost}).$$

5 EXPERIMENTAL SETUP

5.1 Environment

To train an agent, we create a virtual environment composed of two parts. The first part simulates the behavior of a DHW system. We consider an EWH composed of a water buffer of 200 liters and an electrical heating element. When there is a DHW consumption, the hot water is drawn from the buffer and replaced by the same amount of cold water. We model the thermal dynamics with a one-node model as it was done in (Shen et al., 2021). It assumes that water inside the tank is at a single uniform average temperature. The modelling takes into consideration:

- Heat loss from water to its ambient environment that depends on thermal resistance and dimensions of the tank.
- Heat loss due to water demand that depends on the volume of consumed water and on cold water temperature that replaces hot water inside the tank.
- Heat injected inside the tank which depends on the available power to heat the water.

The second part of the environment simulates users' behaviors. We simulate DHW data using (Hendron et al., 2010). The idea is to train an agent on a high number of different DHW consumption scenarios. This can help the agent to extract repeated behaviors, identify probable consumption periods and to adapt hot water production to real needs.

It should be noted that the agent has no access to the described environment. In fact, the modelling is done to create an environment to train the agent and to compute agent's state at each time step.

5.2 Agent Setup

We choose to train a fully connected NN to estimate the action-value vector function with MO-DQN. The size of the output layer of the NN is eight (two objectives and four actions). To test different configurations, we train multiple agents with different preferences over objectives using multiple weight vectors $\mathbf{w}$. Each vector contains, in the following order, a weight for comfort and a weight for energy cost.

Hyperparameters of the NN and the agent were tuned and are shown in Table 1.

Table 1: Hyperparameters of MORL agent training.

Parameter	Value
Memory size (N)	One year
Number of episodes (M)	1000
Episode length (T)	One day
Scalarization function	Linear scalarization
Exploration	Linear decay
Update frequency (K)	Five episodes
Discount factor (γ)	0.95
Number of hidden layers	2
Activation function	Leaky ReLU
Number of nodes	128
Batch size (B)	32
Learning rate (α)	0.0001

5.3 Evaluation Approach

To evaluate the performance of the described method in Section 3 on DHW production problem, we compare it to a conventional rule-based control method. The rule-based method switches hot water production on whenever water temperature is below a threshold T_{min} and is stops when temperature exceeds an upper threshold T_{max}.

We choose to compare multiple MO-DQN agents with different preferences over objectives to rule-based method with different thresholds:

- $T_{max} = 65\ ^{\circ}C$ and $T_{min} = 62\ ^{\circ}C$ (baseline).
- $T_{max} = 60\ ^{\circ}C$ and $T_{min} = 57\ ^{\circ}C$.
- $T_{max} = 55\ ^{\circ}C$ and $T_{min} = 52\ ^{\circ}C$.

Performances are compared on comfort and on energy cost reduction as these are the initial objectives to optimize. Comfort is defined as the proportion of time with a temperature greater or equal than T_{pref} while energy cost reduction is the reduction of cost compared to the baseline. For safety issues, DHW production stops automatically when water temperature is above 65 $^{\circ}C$ for all control methods.

Both rule-based method and MO-DQN are tested and used to produce DHW for unseen consumption data during twelve weeks. The DHW consumption comes from five different domestic water heaters which were measured and made available by (Booysen et al., 2019)

6 RESULTS AND DISCUSSION

Figure 3 shows average results on comfort and energy cost reduction using MO-DQN and rule based method. It appears that minimizing energy cost and maximizing comfort are two conflicting objectives, since maximizing one leads to minimizing the other. In addition, no agent outperforms other agents over both objectives. In other words, all policies learned by agents are incomparable and could be a part of the Pareto front.

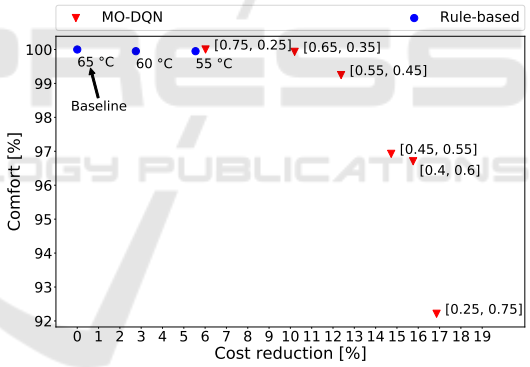
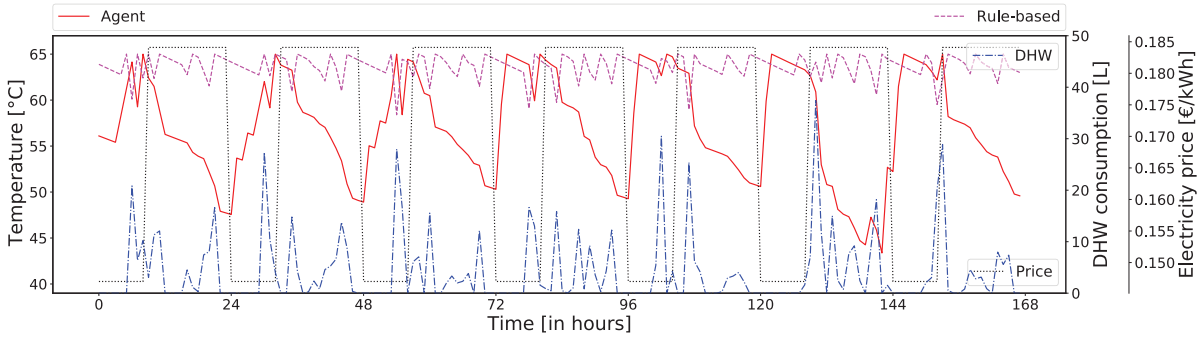


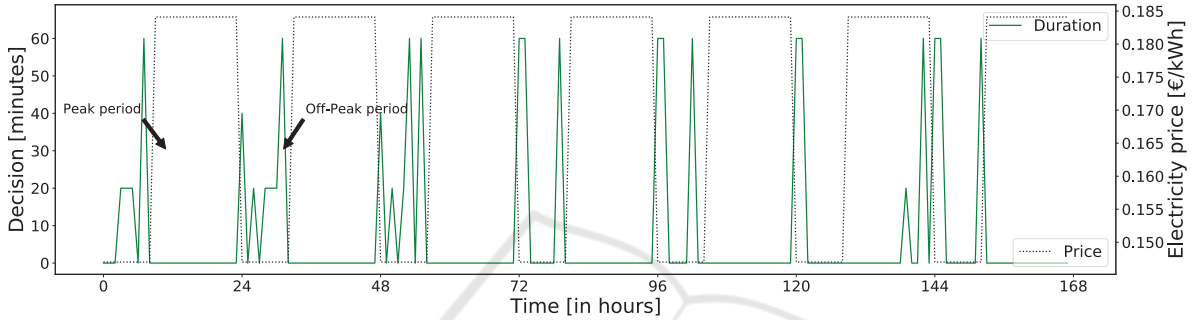
Figure 3: Average results obtained on comfort and energy cost using MO-DQN agents and rule-based method.

Results also show that MO-DQN agents outperform rule-based method with any chosen threshold in terms of cost reduction. Agents offer multiple possible trade-offs between comfort and energy cost. For example, a cautious policy can reduce energy cost up to 10.82% (10.42% on average) without any real impact on comfort (99.9 % on average) when $\mathbf{w} = [0.65, 0.35]$. Other less cautious policies can reach 18% of energy cost reduction on some consumption profiles with a slight impact on comfort.

Table 2 details how agents reduce energy cost according to preferences and focuses on the impact of agents' behaviors on discomfort. Unlike comfort (see definition in Subsection 5.3), discomfort measures the



(a) Temperature profiles in terms of electricity prices and DHW consumption.



(b) MO-DQN agent decisions in terms of electricity prices.

Figure 4: Comparison of DHW production between rule-based method and MO-DQN during one week with $\mathbf{w} = [0.45, 0.55]$.

Table 2: Comparison between different agents to control DHW production. Shown scores are averages obtained on five consumption profiles.

Method	Discomfort	Energy saving (%)	Off-peak actions (%)
Baseline (65 °C)	(0, -)	-	-
$w = [0.75, 0.25]$	(0, -)	2.03	40.76
$w = [0.65, 0.35]$	(1, 38.85 °C)	3.52	57.21
$w = [0.55, 0.45]$	(8, 37.61 °C)	7.35	68.46
$w = [0.45, 0.55]$	(27.2, 37.73 °C)	7.1	72.65
$w = [0.4, 0.6]$	(40, 37.65 °C)	7.78	79.88
$w = [0.25, 0.75]$	(99.4, 37.3 °C)	8.74	78.98

impact on consumption habits and is measured using:

- number of events of DHW consumption with a temperature lower than T_{pref}, and
- average temperature during these events.

It can be noticed that agents minimize energy cost by decreasing energy consumption and/or by shifting DHW production to off-peak periods. These behaviors can expose users to discomfort situations with DHW supplied at a lower temperature than T_{pref}.

Finally, Fig. 4a shows an example of DHW production and compares temperature profiles using MO-DQN and the baseline. MO-DQN agent increases DHW temperature during off-peak periods to be prepared for future DHW consumption. Moreover, temperatures are simply kept above T_{pref} during peak periods to minimize energy cost without minimizing comfort. On the other hand, rule-based method has higher temperature profiles all the time. Figure 4b highlights the link between energy prices and decisions made by the agent. The agent reduces energy cost by shifting DHW production to off-peak periods and by consuming for short duration during peak periods. In summary, the agent tries to produce the needed amount of DHW during off-peak periods and adjusts temperatures according to the demand during peak periods when needs are higher than expected.

These results depend on the modelling described in Section 5.1. In fact, multiple parameters like thermal resistance of the buffer, cold water temperature and available power to heat the water are supposed to be invariant.

7 CONCLUSION

This paper presents MO-DQN, an adaptation of DQN to multi-objective sequential decision problems, The proposed adaptation was designed and applied to con-

trol an EWH in order to maximize comfort and to minimize energy cost. Results showed that the formulation of DHW production as a multi-objective sequential decision problem allows to have multiple policies that can suit each user in terms of preferences. The proposed approach can save energy cost up to 10.24 % in a cautious control case without any real impact on comfort. It turns out that a trained agent with the most conservative policy for comfort can have better results in terms of comfort and cost reduction than decreasing the rule-based control by 10 $^{\circ}C$ compared to the baseline. In future work, these results can be compared to a multi-objective optimization with known DHW consumption needs. Thus, the Pareto front can be estimated and this will allow to check the optimality of the obtained policies.

The presented method can also be used to find trade-offs between energy consumption reduction and comfort for multiple applications. This can be useful during the current energy crisis in Europe and allows energy consumption to be reduced without impacting comfort and habits of users.

Some limitations of the proposed method are known. The method requires a prior knowledge of preferences over different objectives and the expression of preferences can be limited to linear scalarization. In addition, the architecture of the NN can be improved to solve problems with more objectives.

ACKNOWLEDGEMENTS

The authors would thank the partners of the CORENSTOCK Industrial Research Chair, as a national ANR project for providing the context of this work.

REFERENCES

Amasyali, K., Munk, J., Kurte, K., Kuruganti, T., and Zandi, H. (2021). Deep reinforcement learning for autonomous water heater control. *Buildings*, 11(11):548.

Booysen, M., Engelbrecht, J., Ritchie, M., Apperley, M., and Cloete, A. (2019). How much energy can optimal control of domestic water heating save? *Energy for Sustainable Development*, 51:73–85.

Heidari, A., Maréchal, F., and Khovalyg, D. (2022). An occupant-centric control framework for balancing comfort, energy use and hygiene in hot water systems: A model-free reinforcement learning approach. *Applied Energy*, 312:118833.

Heidari, A., Olsen, N., Mermod, P., Alahi, A., and Khovalyg, D. (2021). Adaptive hot water production based on supervised learning. *Sustainable Cities and Society*, 66:102625.

Hendron, B., Burch, J., and Barker, G. (2010). Tool for generating realistic residential hot water event schedules. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).

Hwang, C.-L. and Masud, A. S. M. (2012). *Multiple objective decision making—methods and applications: a state-of-the-art survey*, volume 164. Springer Science & Business Media.

Issabekov, R. and Vamplew, P. (2012). An empirical comparison of two common multiobjective reinforcement learning algorithms. In *Australasian Joint Conference on Artificial Intelligence*, pages 626–636. Springer.

Kapsalis, V., Safouri, G., and Hadellis, L. (2018). Cost/comfort-oriented optimization algorithm for operation scheduling of electric water heaters under dynamic pricing. *Journal of cleaner production*, 198:1053–1065.

Kazmi, H., Mehmood, F., Lodeweyckx, S., and Driesen, J. (2018). Gigawatt-hour scale savings on a budget of zero: Deep reinforcement learning based optimal control of hot water systems. *Energy*, 144:159–168.

Liu, C., Xu, X., and Hu, D. (2014). Multiobjective reinforcement learning: A comprehensive overview. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(3):385–398.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Patyn, C., Peirelinck, T., Deconinck, G., and Nowe, A. (2018). Intelligent electric water heater control with varying state information. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6. IEEE.

Roijers, D. M., Vamplew, P., Whiteson, S., and Dazeley, R. (2013). A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113.

Ruelens, F., Claessens, B. J., Quaiyum, S., De Schutter, B., Babuška, R., and Belmans, R. (2016). Reinforcement learning applied to an electric water heater: From theory to practice. *IEEE Transactions on Smart Grid*, 9(4):3792–3800.

Shen, G., Lee, Z. E., Amadeh, A., and Zhang, K. M. (2021). A data-driven electric water heater scheduling and control system. *Energy and Buildings*, 242:110924.

Tesauro, G., Das, R., Chan, H., Kephart, J., Levine, D., Rawson, F., and Lefurgy, C. (2007). Managing power consumption and performance of computing systems using reinforcement learning. *Advances in neural information processing systems*, 20.

Vamplew, P., Dazeley, R., Berry, A., Issabekov, R., and Dekker, E. (2011). Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine learning*, 84(1):51–80.

Van Moffaert, K. (2016). *Multi-criteria reinforcement learning for sequential decision making problems*. PhD thesis, Ph. D. thesis, Vrije Universiteit Brussel.

Van Moffaert, K., Drugan, M. M., and Nowé, A. (2013). Scalarized multi-objective reinforcement learning: Novel design techniques. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 191–199. IEEE.