Using reinforcement learning to optimize occupant comfort and energy usage in HVAC systems

Article *in* Journal of Ambient Intelligence and Smart Environments · January 2014

DOI: 10.3233/AIS-140288

4 authors, including:

Pedro Viçoso Fazenda
Instituto Politécnico de Lisboa
10 PUBLICATIONS **78** CITATIONS
SEE PROFILE

Pedro U. Lima
University of Lisbon
283 PUBLICATIONS **2,700** CITATIONS
SEE PROFILE

Journal of Ambient Intelligence and Smart Environments 0 (2014) 1–0
IOS Press

Using Reinforcement Learning to Optimize Occupant Comfort and Energy Usage in HVAC Systems

Pedro Fazenda [a,b,*], Kalyan Veeramachaneni [b] Pedro Lima [a] Una-May O'Reilly [b]
[a] *Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa*
[b] *Computer Science and Artificial Intelligence Laboratory, MIT, Boston, USA*

Abstract. The present paper suggests a procedure to enhance the operation of the heating, ventilation and air conditioning system, following the idea that a multi-objective optimal supervisory control for such a system should consider the cost of energy, activity schedules, occupancy patterns and the individual comfort preferences of each tenant. Considering that tenants tend to forget to adjust systems appropriately and that, in many spaces, the conditioning requirements are not adjusted to the occupancy of those spaces, the result is unnecessary energy waste. This paper studies the application of a discrete and a continuous reinforcement-learning-based supervisory control approach, which actively learns how to appropriately schedule thermostat temperature setpoints. The result is a learning controller that learns the statistical regularities in the tenant's behavior, allowing him/her to meet comfort requirements and optimize energy costs. Results are presented for a simulated thermal zone and tenant.

Keywords: HVAC, Ambient Intelligence, Reinforcement Learning

1. Introduction

Buildings are responsible for a significant amount of the global energy usage [1,2]. In the past few years, research communities have been addressing the energy efficiency challenge for building sustainability in several different but interdependent areas. The artificial intelligence (AI) community, for instance, has been actively engaged in creating smart environments for Smart buildings (SBs) [3,4,5]. The vision for SBs includes having a building management system (BMS) that incorporates the best available concepts and state-of-the-art technologies to automate and optimize the operation of integrated systems and services.

A Smart BMS must provide an environment with some sort of ambient intelligence that anticipates the needs of the tenants, by learning their time-variant preferences and habits, in an unobtrusive and transparent way [6]. It should take actions towards minimizing energy wastage and operating costs, by subtly manipulating appropriate controls which are internally regarded as design parameters of a multi-objective optimization problem. The algorithm goals are to minimize the energy wastage without sacrificing discernible tenant comfort. This constraint implies that the algorithm should avoid taking actions that are likely to cause discomfort to the tenants.

The exemplified smart building management service in this paper is the building's heating, ventilation and air conditioning (HVAC) system. This system is one of the most energy demanding systems inside a building. It includes the equipment, distribution network and terminals used either collectively or individually to provide fresh filtered air, heating, cooling and humidity control in a building. Space heating and cooling can take up to 40% of the final energy in residential buildings and 20% in commercial buildings [7]. Energy can be saved by systems with advanced controls and algorithms which perform real-time optimization of operation parameters.

*C. author E-mail: copytopedro@gmail.com.

Optimizing the operation of the HVAC, in many current buildings, includes strategies such as setting the operation of the system to a *low power* state, according to a certain schedule. This means that comfort requirements are not guaranteed in unscheduled hours, when there is low occupancy (for example, late classes or meetings). On the other hand, during normal operating hours, energy is unnecessarily wasted in many situations. Tenants tend to forget the HVAC *on* when they leave their rooms, and heating and cooling loads are set to guarantee certain fixed setpoints, instead of being intelligently exploited to save energy.

To go beyond simple automation strategies for intelligent HVAC control, a BMS should learn the tenant's preferences by observing behavior, and perform according to those preferences. To minimize the amount of energy used the HVAC should be turned *off* if the zone is expected to be unoccupied, or if there is any other cost effective means to guarantee the same comfort levels.

1.1. System description

Addressing the requirement that a smart BMS should adapt its operation according to the cost of energy and comfort level of its tenants, this paper proposes two reinforcement learning (RL) strategies for two different problems. The first problem, which is called *Bang-Bang Heater*, presumes that a heating unit is controlled at a low level by a controller that guarantees that the environment temperature will converge to a certain setpoint, when the heater is *on*. The heater can either be turned *on* or *off* but does not have an interface for the BMS to set the temperature setpoint. The second problem, called the *Setpoint Heater*, presumes the HVAC unit has a temperature setpoint control interface.

Both the *Bang-Bang Heater* and *Setpoint Heater* problems are applied to the a thermal zone system represented in Figure 1. A tenant can express dissatisfaction with the current temperature by pressing the "UP" or "DOWN" arrow on the thermostat indicating that the temperature should be increased or decreased. With these two elements of information, over repeated interactions, the controller must adjust zonal temperature efficiently - to just the right temperature and with a maximum of energy savings.

The smart BMS control algorithm should learn when to turn the heater *on*/*off* (*Bang-Bang Heater*) or set the temperatures setpoints (*Setpoint Heater*) to appropriate values throughout the day. This control must be executed in a manner that minimizes the number of tenant signals and at the same time also minimizes the costs associated with heating and cooling. Even when the tenant forgets to reset a temperature for zonal vacancy, the smart BMS must discover and exploit the opportunity of the tenant's absence, or satisfaction with current environmental conditions (inferred through the lack of action over the control interface), for more efficient energy management.

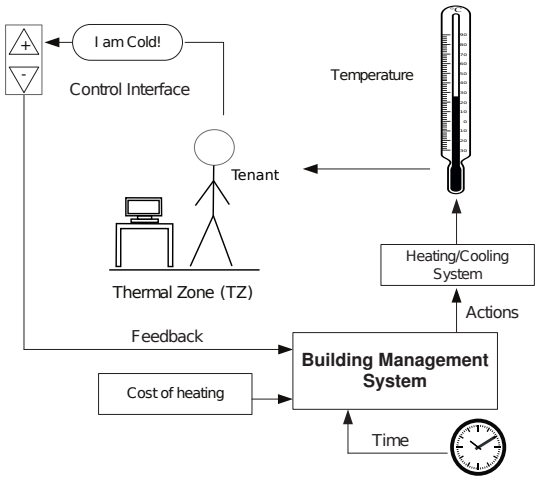

Fig. 1. The tenant indicates if s/he is feeling cold or hot. The smart Building Management System must learn how to respond helpfully to the tenant by minimizing the required number of thermostat interactions while saving energy.

The *Bang-Bang Heater* is solved with the well known Q-learning algorithm [8], an RL algorithm [9,10,11,12,13], with straightforward discrete states and actions. The *Setpoint Heater* is solved with a continuous state and action Q-learning algorithm, via a method called *wire fitting* [14].

Both algorithms are demonstrated by setting up a simulated environment with simulated tenant behavior and low level (bang-bang or setpoint) heater control behavior. It is assumed that the external temperature is colder than a comfortable zone temperature when it is occupied. The simulation components are connected to each Q-learning algorithm which acts as a smart thermal controller.

The remainder of the article is organized as follows. Related work is presented in Section 2. The optimization approach presented in this paper employs Q-learning RL algorithms. Therefore, Section 3 provides a brief but necessary technical introduction to Markov decision processes, RL and sequential decision problems, including the *wire fitting* method which extends

Q-learning to problems with continuous states and actions. Section 4 presents results of computational experiments running the algorithms with a single simulated tenant and thermal zone. Section 5 discusses results and Section 6 presents the conclusion and future work.

2. Related Work

In the last three decades a considerable amount of work has been done on using building automation strategies for energy savings, which naturally includes optimizing the operation of systems such as the HVAC. Dounis and Caraiscos [15] present a survey with the state of the art on control systems for energy and comfort management in buildings. Work has spanned across multiples strategies. Control techniques, in general, are categorized into hard control and soft controls [16]. We are interested in soft control techniques, where most AI algorithms are included. Most AI techniques have somehow been applied in building control strategies. This includes Evolutionary algorithms [17,18,19,20], Neural networks [21,22,23], Bayesian networks [24] and other algorithms for control and optimization [25,26,27,28].

To express tenants' preferences using linguistic labels that human operators can understand, a popular approach for HVAC control has been to use fuzzy controllers [29,18,17]. Fuzzy logic is useful in capturing and representing imprecise notions such as "hot" and "very hot". However, it has several limitations [30,31]. The fuzzy part is hard to program and prior knowledge is required to model the fuzzy system. The knowledge base is usually constructed based on operators' experience and requires fine tuning and simulation before becoming operational. Their knowledge is often incomplete and episodic, rather than systematic.

The advantages of using a RL approach to machine learning, as an option over other paradigms, are well known [32,33]. RL can applied in situations where tenants do not know the correct answers required for supervised learning. The system is expected to learn how to achieve goals, by trial and error, in real time, with continual feedback from its environment. This real-time, closed-loop, goal-seeking behavior seems to be a crucial aspect of how humans operate, and is an interesting paradigm to be explored in SBs. Some of the unknown answers in the environment include occupancy patterns, thermal preferences and how variables such as temperature will evolve inside building spaces.

Models for Comfort and Building Systems

Many authors use predefined models that represent expected behaviors for tenants and building systems. Although these models are used for decision support in algorithms, they are not always fully adjusted to reality. For example, Meyer and Emery [34] propose an air conditioning system controller that generates an optimal plan to use thermal energy storage in the buildings, by shifting part of the daily cooling loads to the night off-peak hours, when electricity prices are lower (while considering inputs such as weather forecast and indoor heat gains). All parameters associated with the HVAC system and the building thermal response function have to be previously characterized, and authors suggest a procedure to identify system components. But the resulting models are highly prone to noise and modeling errors (e.g. opened windows influence the thermal storage of a building) and most characteristics are not time invariant. They degrade over time and depend, in many situations (e.g. example the building structure) on environmental conditions like e.g. temperature, humidity, etc.

Another very common approach has to do with using models for thermal comfort. For example, Dalamagkidis and Kolokotsa [35] developed an environmental controller, using RL, that sets the cooling/heating level and opens/closes a window, by following a policy that maximizes a reward function based on a fixed weighted average of three factors: energy used, comfort and air quality. To estimate thermal comfort, they employ the predictive mean vote (PMV)/ predicted percentage of dissatisfied (PPD) (Fanger's comfort model)[36,37]. However, many researchers showed the limitations of the PMV/PPD model [38,39,40]. The derived comfort equations are based on standardized assumptions such as clothing, air velocity, activities, etc. The algorithms that use these models may converge to temperature values that may not be optimal at all times. Adaptive comfort [41] is based on the assumption that the comfort perception of people will depend on the outdoor climate conditions.

According to Mathews et al. [42], "For low outside air temperatures, people will be comfortable if the indoor temperature is lower. The opposite is true for high outside air temperatures. This affords us the opportunity to potentially save even more energy. If the indoor temperature can be cooler when the outside air temperature is low, less heating would be required. During periods of high outside air temperatures, a higher temperature would imply that less cooling would be

required" (p. 153). Williamson and Riordan [43] point out the fact that "...the reaction of people to a sense of being cold or hot is not necessarily to operate a heater or cooler, nor is such a reaction generally the sole response. Adjusting clothing, altering activity levels etc. are also common responses." (p. 1). This affords us the opportunity to potentially save even more energy. SBs should include environment controllers with the ability to learn and self-regulate according to the thermal preferences of tenants. They need to learn rules of behavior based on feedback they obtain from occupants, and continually adapt this knowledge [44].

Scheduling

The simplest strategy to save energy with the HVAC includes using thermostats. A thermostat acts as an interface between the tenants' thermal preferences and the operation of the heating and cooling systems by maintaining the temperature near a desired setpoint. They can range from simple mechanical control mechanisms to internet enabled programmable devices [45, 46], offering programming options which allow tenants to define several setpoints according to a schedule. It can also display information like the current temperature, ventilation rate and more recently, with smart metering, pricing feedback. Unfortunately, it has been shown that many tenants do not explore most functions of these interfaces due to the fact that they do not understand them [47]. Even for people who understand these interfaces, programming setback schedules for every day of the week and time of year is a tedious task.

Our priority is to avoid demanding the tenant to operate any temperature controller more complicated than the simple and inexpensive thermostat which is standard in most buildings – one that allows the tenant to command the temperature to be increased or decreased. Such a thermostat may optionally provide a temperature gauge but we even assume such a feature is unnecessary. We believe that tenants can be perfectly satisfied with a simple interface if their inputs, using that interface, result in temperature setpoints that comply with their comfort requirements. Our motive is to effectively circumvent sophisticated thermostats that require even small amounts of set up or direct use because these features are frequently ignored and wasted by tenants.

Building administrators are also satisfied if the BMS can automatically and dynamically adjust the HVAC operation to optimual schedules. To circumvent the task of programming schedules, some authors have tried to estimate activities and occupancy by observing the environment. This include, for example, observing CO_2 levels [48], monitoring the electrical load of the house and the hot water heating pattern over a certain period [49], or even using smart-phones as tenant-location devices, to predict the arrival times of tenants and modifying temperature setpoints accordingly [50]. This information may be useful for decision support but, in the end, the problem is reduced to learning how to operate within maximum energy efficiency, while trying to minimize the number of "complaints".

The controller of this paper explores the heating setpoints that satisfy tenant comfort, while minimizing the needs for thermal energy. No modeling is needed neither for modeling comfort nor building structures and components. All the necessary information is extracted from the tenant interaction and the cost associated with the conditioning of the environment. We use RL (Q-learning) and consider the interaction of the tenant and the energy used for heating and cooling in the reinforcement signal. The BMS is penalized in proportion to the amount of energy that it uses to guarantee the current temperature setpoint, including an additional penalization if a tenant acts on the interface at any instance. We propose a solution where the actions can be continuous (manipulating the temperature setpoints) and discrete (turning a heating/cooling system *on*/*off*).

3. Technical Approach

The central control process of the proposed solution starts by the algorithm taking control actions that change the zone temperature. The BMS interacts with the environment. It needs to decide how to change the temperature setpoint before waiting for a new state observation and feedback, as illustrated in Figure 2.

The control problem is modeled as a *Markov Decision Process* (MDP)[13] in a fully observable environment with a Markovian transition model and additive rewards. The process is formally represented as a 4-tuple (S, A, P, R), where the environment transitions through a series of states s_t from a set of finite possible states S. In our particular problem, time is one of the state components and, at each time step, the agent selects an action a_t from a finite set of possible actions A. As a result the environment's state is updated to s_{t+1} and the agent receives a reward r_t from a set of rewards R. These rewards are real-valued and can be positive

or negative, but must be bounded. The descriptor "fully observable" implies that all possible states of the environment are detectable by the agent. The Markovian transition model implies that the probability of transitioning from state s to state s', after executing action $a \in A$, depends only on the current state and not on the history of earlier states.

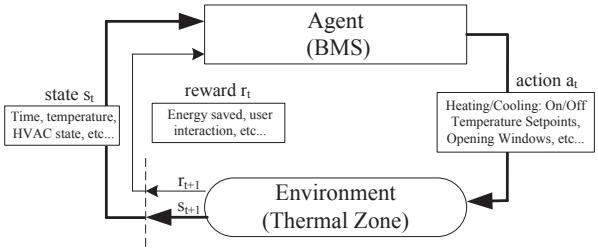

Fig. 2. The interaction between an agent and its environment. In the problem formulation, the environment is the thermal zone and the agent is the BMS. Actions are temperature setpoint changes by the control algorithm and rewards (negated as penalties) are subsequent tenant signals in the form "I am cold" or "I am hot", plus the cost of energy used.

A *policy* $\pi(s)$ serves to define what an agent should do for any state that might be reached. The purpose of the learning agent is finding the optimal policy, i.e., selecting its actions to maximize its expected utility (discounted sum of expected future rewards) when π is followed starting from s, given by:

$$U^{\pi}(s) = E\{r_0 + \gamma r_1 + \ldots | s_0 = s; \pi\} \qquad (1)$$

where $\gamma \in [0, 1]$ is the discount factor that describes the preference of an agent for current rewards over future rewards. When γ is close to 0 rewards in the distant future are viewed as insignificant. When γ is 1, discounted rewards are exactly equivalent to additive rewards.

In this paper the reward is inverted to a penalty. The controller seeks to select its actions in order to minimize the cost of the energy it uses and the number of times a tenant subsequently interacts with the thermostat.

3.1. Q-Learning

Q-learning [8] is a RL technique that is used to find an optimal action-selection policy for each state. It is a provably optimal algorithm under the theoretical assumptions of infinite learning time. The value of doing an action a in state s is denoted by $Q(s, a)$ and these Q-values are directly related to utility values as follows:

$$U(s) = \max_a Q(s, a)$$

The Q-learning agent uses the 1-step temporal-difference TD($\lambda = 0$) [51,13] as its estimator of expected returns. At each time step, Q-values are adjusted according to following update:

$$\Delta Q(s, a) = \delta(R(s) + \gamma \max_{a'} Q(s', a') - Q(s, a))$$

$$Q(s, a) \leftarrow Q(s, a) + \Delta Q(s, a) \qquad (2)$$

which is calculated whenever action a is executed in state s leading to state s'. Parameter δ is the learning rate $(0 < \delta \leq 1)$ that controls convergence to optimal action-values Q^*. If each action is executed in each state an infinite number of times, and δ is decreased with an appropriate schedule, the algorithm converges.

When the learning process starts, the controller has no previous knowledge of what actions to select. Its strategy is to explore its options while exploiting whatever feedback, with respect to its objectives, it can get. The Q-learning agent has to balance between the decisions of following a known policy with the need to further explore the state-space in order to find a better policy that will bring higher rewards. This is useful because the environment model can change and can become unadjusted to the learned policy. An agent therefore must make a trade-off between exploitation to maximize its reward as reflected in its current utility estimates and exploration to maximize its long-term performance. There are many schemes on how the agent balances this decision e.g., the agent may choose a random action in some instances (depending on and exploration rate parameter) and follow a *greedy* policy (best action)[52] otherwise.

3.2. Continuous State and Action Q-Learning

In Q-learning, policies and value function are implemented using a two-dimensional lookup table indexed by state-action pairs. Thus it deals solely with a discrete and finite number of states and actions. The *Setpoint Heater* controller assumes that actions and states are continuous values: the temperature is controlled (or changes) smoothly in minutely small measures and zonal temperature is effectively continuous. Scaling to

large numbers of states and actions as a means of covering the range of continuous states and actions is impractical because of computational complexity. Generalization between states and/or actions can be introduced to Q-Learning by using function approximation instead of table-based storage.

To deal with high-dimensional continuous states and actions we used the *wire fitting* method proposed by Baird and Klopf [14] and used by Gaskett et al. [53]. It is a continuous state, continuous action q-learning method where actions vary smoothly with smooth changes in state. Actions are quickly generated by using a function approximation system, e.g. an artificial feedforward neural network (NN), to map the state s into a set of action-value pairs (a_i, q_i), called *wires*, as illustrated in Figure 3.

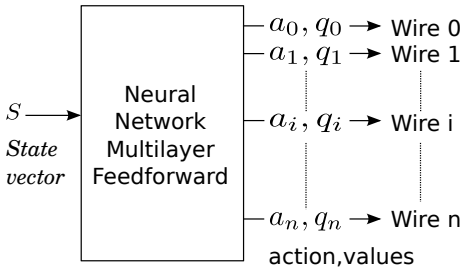

Fig. 3. Using an artificial neural network to map the state into a set of action-value pairs (Wires) (adapted from [53]).

The best action (q-value given by: $\max_i q_i(s)$) can be immediately selected from the output of the NN. The generalization to other action values other than the ones given by the set of wires is accomplished by using a *wire fitting interpolation* function as shown in Figure 4.

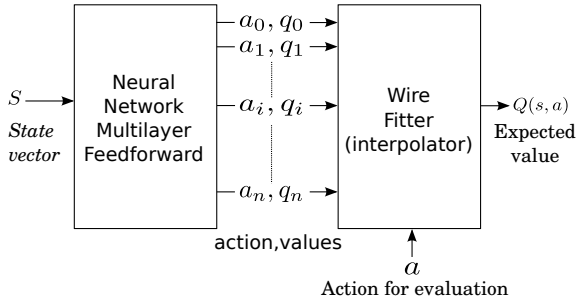

Fig. 4. Wire fitted neural network architecture (adapted from [53]).

The *wire fitted neural network* architecture also uses this interpolator to train the outputs of the NN. The locally weighted interpolation function is given by (3), where action a and state s are vectors, possibly with a different number of elements; i is the wire number, n is the total number of wires; $a_i(s)$ is the ith action vector and $q_i(s)$ the corresponding q-value; c is a "smoothing" factor [1] and ε avoids division by zero.

The interpolator defines $Q(s,a)$, for a particular a, to be a weighted average of q_i values. If the action for evaluation is near a particular a_i, then the corresponding q_i is given more weight. The action for evaluation is usually the action with the highest q-value when following a greedy policy. Thus the interpolator output will be the q-value of the wire associated with this action.

Figure 5 shows the wire fitting process with three wires placed at:

$$\{(0.2, 0.3),(0.5, 0.7),\ (0.8, 0.5)\}$$

for $c = 0.5$, $c = 0.0$ and $\varepsilon = 0.001$.

$$Q(s,a) = \lim_{\varepsilon \to 0} \frac{wsum(s,a)}{norm(s,a)} \tag{3}$$

with,

$$wsum(s,a) = \sum_{i=0}^{n} \frac{q_i(s)}{dist_i(s,a)}$$

$$norm(s,a) = \sum_{i=0}^{n} \frac{1}{dist_i(s,a)}$$

and

$$dist_i(s,a) = \|a - a_i(s)\|^2 + c(\max_i q_i(s) - q_i(s)) + \varepsilon$$

A property of the wire fitting interpolator is that the highest interpolated value always coincides with the interpolation point defined by the wire with highest Q-value. (3) is a continuous, smooth function of its inputs so it is possible to back propagate errors through the wire-fitting block to update weights in the function approximation system [54] (in this case the NN).

The update $\Delta Q(s,a)$, defined by (2), result in the necessary adjustments that need to be made to each wire, as the example shown in Figure 6. This adjustment, represented in Figure 7, is accomplished by training the NN through backpropagation of errors.

[1] To simplify the interpretation of (3), consider c = 0.

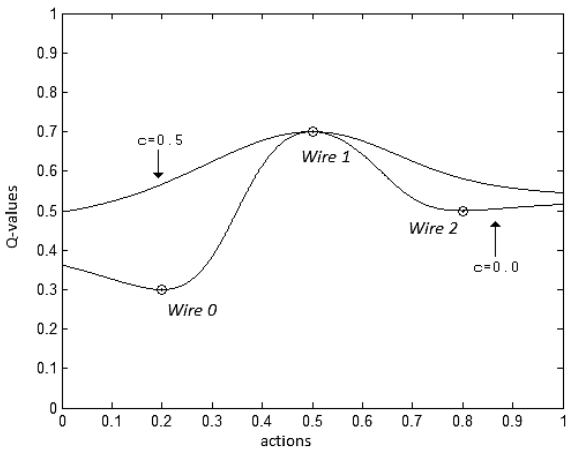

Fig. 5. Weighted-nearest-neighbor interpolation of the three control points (shown as ∘) for $c = 0.5$ and $c = 0.0$ (smoothing factor).

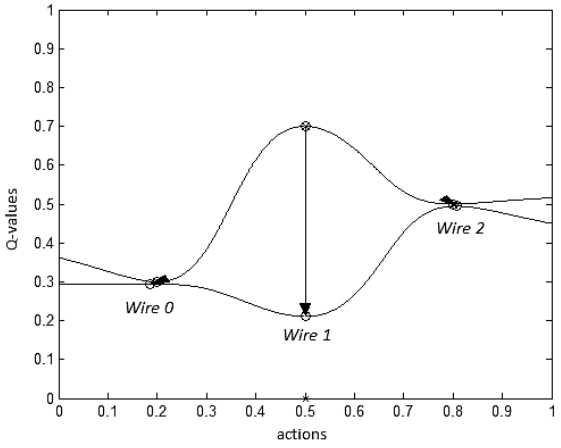

Fig. 6. Updates made to the wires (shown by the arrows). Selected action: $a = 0.5$, $c = 0.0$, $\Delta Q(s, a) = -0.5$

4. Experimental Simulation

This section presents a set of simulations executed in Matlab. In these experiments a policy for an agent (the BMS) to act on the HVAC system is obtained, in accordance to specific requirements that are reflected through the reinforcement function R. The experiments are discrete time event simulations where both the tenant and thermal zone are modeled. The BMS goes through a learning phase that takes several episodes (e.g. a 24 hour period). Each episode is divided into a number of time steps Δ_t and, at each step, the BMS chooses an action based on its policy that affects the temperature inside the thermal zone. The behavior of the tenant is also simulated by considering

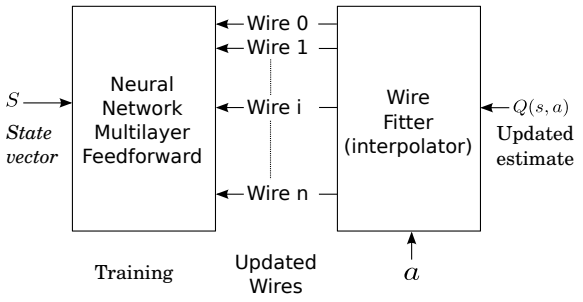

Fig. 7. Wire-fitted neural network training algorithm (adapted from [53]).

that he follows a working schedule and becomes uncomfortable if the temperature is not within a comfort range. Based on the state of the heating system and the action of the tenant, the reinforcement is calculated and used for updating Q-values.

Without loss of generality, we assume the outside temperature is lower than what is desirable if the room is occupied. This implies that the controller's job is to turn on the heating only as frequently as is necessary to make the tenant comfortable. It can neglect heating the room at all when the room is unoccupied but may be required to "preheat".

In the first experiment, heating is supplied by equipment like an electric space heater, that has no thermostatic adjustment (by the BMS) and can only be turned *on* or *off* by, for example, cutting off the power supply. We assume that the tenant will adjust the temperature setpoint to a comfort level and that this temperature is reached and maintained if the heater is left *on* for a certain amount of time. The state of the system contains a variable that defines if the "heater is *on*" or "heater is *off*". The controller actions are to either reverse ("toggle") the heater's status or leave it alone.

In the second experiment, the BMS can control the temperature setpoint by setting the input of an heating controller that can control the temperature of the thermal zone. This controller will guarantee the convergence of the temperature to the values demanded by the BMS.

After the learning phase, a single episode is simulated with the learned policy.

4.1. Simulating the thermal zone

A single thermal zone is considered with the simplified thermal dynamic model given by (4) to (6) and represented in Figure 8. Parameter C_1 represents the thermal capacitance [J °C^{-1}] and R_1 the thermal re-

sistance associated with the boundaries of the thermal zone [°C W^{-1}].

$$\dot{\mathtt{T}}_1(t) = \frac{1}{C_1}[q_i(t) - q_o(t)] \quad (4)$$

$$q_o(t) = \frac{1}{R_1}[\mathtt{T}_a - \mathtt{T}_1(t)] \quad (5)$$

$$\dot{\mathtt{T}}_1(t) + \frac{1}{R_1 C_1}\mathtt{T}_1(t) = \frac{1}{C_1}q_i(t) + \frac{1}{R_1 C_1}\mathtt{T}_a \quad (6)$$

where $\mathtt{T}_1$ and $\mathtt{T}_a$ are the indoor and outdoor ambient

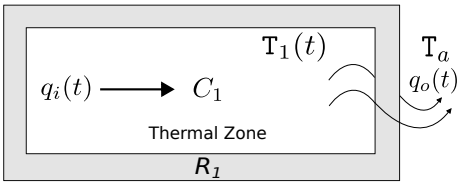

Fig. 8. Simplified thermal dynamic model for the thermal zone.

temperatures [°C] and q_i, q_o [W] represent the input and output heat fluxes.

The stationary indoor temperature is given by $\mathtt{T}_d = \mathtt{T}_a + R_1$, with $q_i(t)$ being the unitary step function. When the system is controlled through an *on/off* interface, we consider that there is a fixed setpoint (comfortable temperature $\mathtt{T}_d$, set by the tenant) and the heating system will drive the indoor temperature from $\mathtt{T}_a$ (heater *off*) to $\mathtt{T}_d$ (heater *on*).

4.2. Simulating the Tenant

The behavior of the tenant is simulated using a finite state machine with the following states: *Out(0)*, *Working(1)*, and *Uncomfortable (2)*. The state transitions between *Out* and *Working* depend on a stochastic schedule that is generated prior to each simulation period. The instants of arrival and departure from the thermal zone are generated from configurable normal distributions. Uncertainty in tenant's thermal preferences can be expressed using a α-*level* fuzzy set of temperature values $\mathtt{T}$ [55]. The desired value or setpoint is chosen as a trapezoidal fuzzy interval whose membership function is illustrated in Figure 9. The trapezoidal type-1 fuzzy set is equipped by an α-*cut*, with $\alpha \in [0, 1]$ being a random variable taken from a uniform distribution to model the uncertainty of comfort.

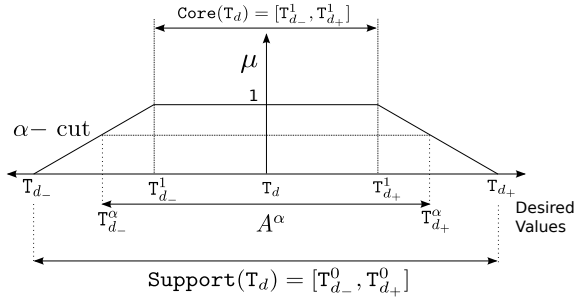

Fig. 9. Trapezoidal fuzzy interval for desired temperature values (adapted from [55]).

$\mathtt{Core}(\mathtt{T}_d) = [\mathtt{T}^1_{d_-}, \mathtt{T}^1_{d_+}]$ is composed of the most acceptable tenant's preferences and $\mathtt{Support}(\mathtt{T}_d) = [\mathtt{T}_{d_-}, \mathtt{T}_{d_+}]$, defines the upper and lower bounds of comfortable temperatures for all values of α. The desired temperature values belong to the fuzzy interval $A^\alpha = \{\mathtt{T} : \mu(\mathtt{T}) \geq \alpha\}$ and the tenant's state transitions from Working to Uncomfortable if the indoor temperature $\mathtt{T}_1 \notin A^\alpha$.

A tenant will show careless behavior towards energy use by always leaving the heating on even when the room is empty. Figure 10 shows the tenant and HVAC state with no BMS actuation and assuming full certainty in the tenant's preferences by setting $\mathtt{T}_d$= 22 °C and $\mathtt{Core}(\mathtt{T}_d) = \mathtt{Support}(\mathtt{T}_d) = [20, 24]$ °C.

The tenant arrives at $t = 80$ and starts working at instant $t = 81$. He becomes uncomfortable at instant $t = 82$ and acts on the HVAC system by switching it *on* (this action is represented in the figure with an asterisk) at that instant. The tenant leaves for lunch between $t = 135$ and $t = 175$ and returns home at $t = 220$. The temperature graph shows the outdoor temperature $\mathtt{T}_a = 18$°C and two lines representing: [1] $\rightarrow$ ($\mathtt{T}_d$); (comfortable temperature) [2] $\rightarrow$ ($\mathtt{T}_{d_-}$) (temperature comfort limit).

4.3. The Bang-Bang heater problem

The Bang-Bang heater problem, represented in Figure 11, is solved using standard Q-learning with discrete states and actions. The state of the system is represented by a vector (t, h), where $t \in \{t_1, t_2, \ldots, t_T\}$ indicates the time the system has been in operation and t_T, the corresponding lifetime desired for the system. $h \in \{0, 1\}$ represents the environment state i.e., the heating is *off* $(h = 0)$ or *on* $(h = 1)$. The system can observe h and has the possibility to act upon this interface.

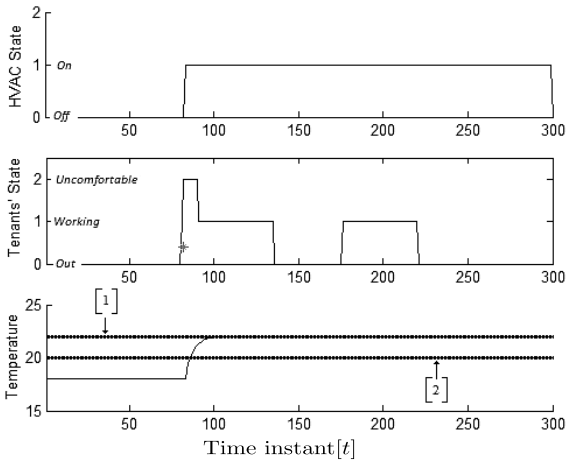

Fig. 10. Tenant and HVAC state with no BMS actuation. The tenant arrives at $t = 80$, $t = 175$ and leaves at $t = 135$, $t = 220$. He is uncomfortable between $t = 81$ and $t = 90$. The HVAC is kept *on* even when the tenant leaves the thermal zone. [1] - Comfortable temperature (22°C), [2] - Temperature comfort limit.

	HVAC State - h	Action - a
Q_{off_M}	*Maintain*	*off*
Q_{on_M}	*Maintain*	*on*
Q_{off_T}	*Toggle*	*off*
Q_{on_T}	*Toggle*	*on*

Table 1

Q-values associated with each HVAC state-action configuration, for each instant t.

Actions $a \in \{0, 1\}$ include the possibility to **Maintain** $(a = 0)$ the state of the system or **Toggle** $(a = 1)$ the current state.

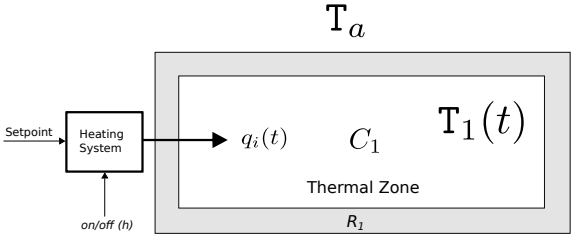

Fig. 11. In the Bang-Bang example the heater has a fixed temperature setpoint (Set by the tenant). The BMS turns the heater *on*/*off*.

The Q-learning BMS stores Q-values in memory. For each instant t we have 4 different state-action configurations as shown in Table 1.

The reinforcement function is given by the following linear combination:

$$R(t, h) = -w_1 interaction(t) - w_2 h$$

where w_1 and w_2 are weights with $w_1 + w_2 = 1$. It regards the usage of the HVAC and the interaction of the tenant with the system. It will penalize states where the heating is *on* $(h = 1)$, or if the tenant has acted upon the system i.e., $interaction(t) = 1$ $(interaction(t) = 0$, otherwise).

The learning phase took 250 iterations (each iteration with a duration $t_T = 300$) with a learning rate $\delta = 0.3$, discount factor $\gamma = 0.91$ and the following weights: $w_1 = 0.9, w_2 = 0.1$. These values were determined empirically.

Figure 12 shows the convergence of the Q-Values and Figure 13 shows the final results of using the learned policy. The BMS tries to minimize the HVAC usage by switching if *off* regularly, but turning it *on* at instances that anticipate the tenants' action. The tenant remains *Uncomfortable* while the temperature is below the comfort limit line but will not act because the HVAC is already *on*.

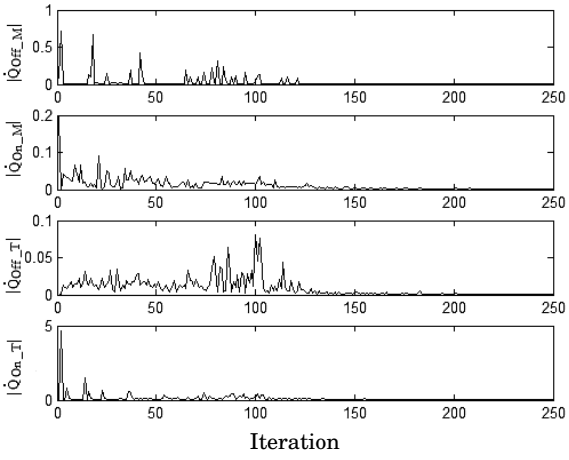

Fig. 12. The convergence of Q-Values: $|\frac{\partial Q}{\partial t}| \rightarrow 0$.

4.4. The Setpoint Heater problem

The Setpoint Heater problem uses continuous state Q-Learning with a wire fitting interpolation function to determine the setpoint temperature at every instant. The state of the system is given by the time vector of the previous example. The action $a \in [0, 1]$ at each instant sets the thermostat setpoint T_p by linear mapping to the following temperature interval: $T_p \in [T_a, T_d]$.

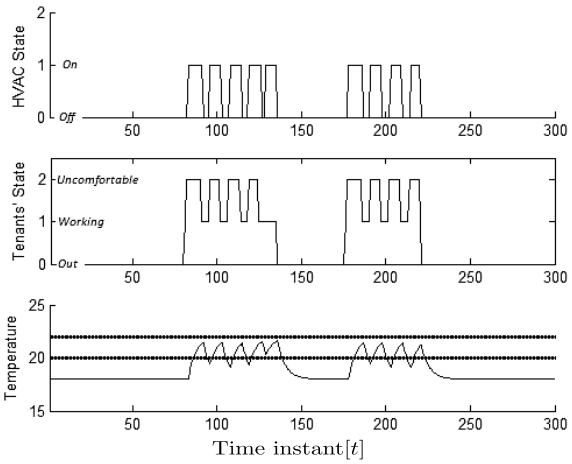

Fig. 13. Policy application results. The BMS learns how to maintain the temperatures within limits that explore the boundaries of comfort, while saving energy when the thermal zone is unoccupied.

Simulations started with three wires, initially placed at:

$$(a_0, q_0) = (0.1, 0.1)$$
$$(a_1, q_1) = (0.5, 0.1)$$
$$(a_2, q_2) = (0.9, 0.1)$$

The wires are updated during a learning interval $\{t | t \in \mathbb{R}, 0 \leq t < t_T\}$, according to the following reinforcement function:

$$R(t, \mathtt{T}_p) = -w_1 hC(\mathtt{T}_p) - w_2 interation(t) cC(\mathtt{T}_p)$$

where hC and cC are heating and comfort cost functions given by (7) and (8). The comfort cost function serves as an heuristic to guide the search towards comfortable temperatures when the tenant is interacts with the thermostat.

$$hC(\mathtt{T}_p) = \frac{1}{\mathtt{T}_d - \mathtt{T}_a}(\mathtt{T}_p - \mathtt{T}_a) \tag{7}$$

$$cC(\mathtt{T}_p) = -\frac{1}{\mathtt{T}_d - \mathtt{T}_a}(\mathtt{T}_p - \mathtt{T}_a) + 1 \tag{8}$$

Figure 14 shows the final results after a 80 day learning period using $w_1 = 0.45$ and $w_2 = 0.55$, discount factor $\gamma = 0.8$, learning rate $\delta = 0.9$, "smoothing" factor of the wire fitter $c = 0.0$ and *greedy* selection strategy. Figure 15 shows the average daily reinforcement $\hat{R}$ received over the learning period.

The system learns the occupancy pattern and tries to minimize the supply of heat. Due to the lack of interactions by part of the tenant, the BMS learns how to let the temperature go down in the intervals when the thermal zone is unoccupied. In some situations, like the one presented in the example, there are periods when the temperature levels are uncomfortable to the tenant.

Figure 16 shows the resulting simulation when more emphasis is given to tenant comfort. With weights set to $w_1 = 0.1$ and $w_2 = 0.9$ ceteris paribus, the system maintains the temperature at levels that are comfortable to the tenant (values of $\mathtt{T}_1$ which he normally never acts) in an interval that covers any period that the tenant might be in the zone, including the interval when he is out to lunch.

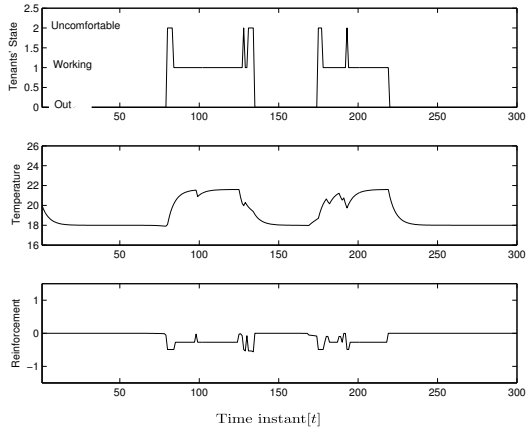

Fig. 14. Setting the temperatures according to the occupancy pattern and maximizing for energy efficiency. The tenant becomes *Uncomfortable* in some situations.

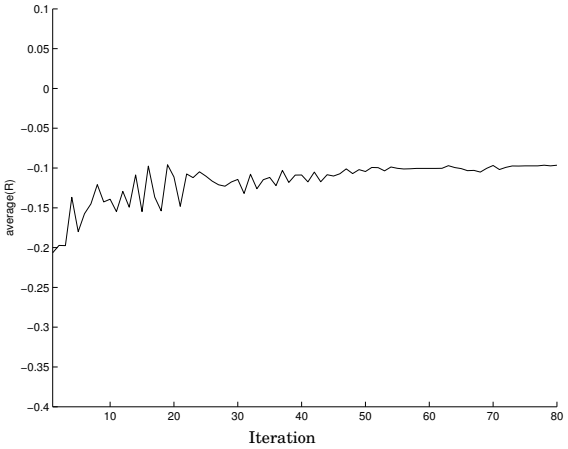

Fig. 15. Average daily reinforcement $\hat{R}$ received over a learning period of 80 days.

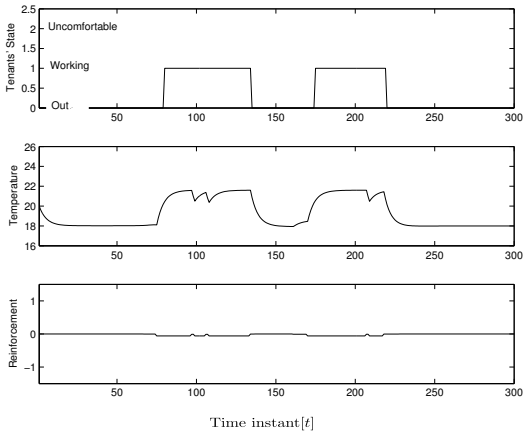

Fig. 16. Setting the temperatures according to the occupancy pattern while giving more emphasis to tenant comfort.

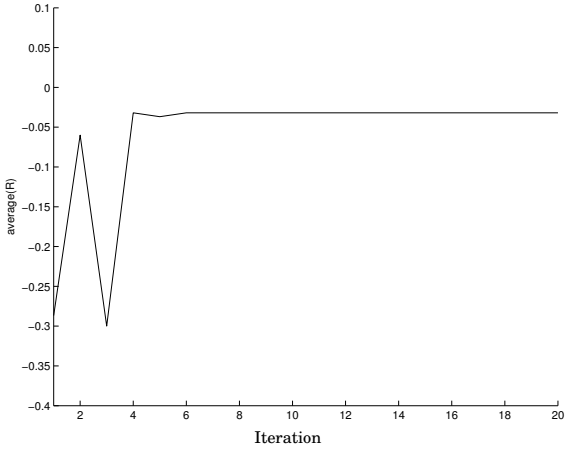

Fig. 17. Average daily reinforcement $\hat{R}$ using a wire table.

4.4.1. Discrete Time

If time can be discretized in an application, as the example used in the *Bang-Bang Heater* heater problem where $t \in \{t_1, t_2, \ldots, t_T\}$, the NN used in the wire fitting method as a function approximation system can be replaced by a table containing T sets of wires. Using a table to store the state of the wires can hold significant improvements to convergence of the algorithm by avoiding fitting errors. We observed that these errors penalize the quality of the solutions and the learning period. Figure 17 shows the average reinforcement when using a wire table. The algorithm converges in just 4 days.

The resulting temperatures of this example, represented in figure 18, also show the extreme situation of using a discount factor of $\gamma = 0.0$. By fully maximizing its immediate rewards, in detriment of any future rewards, the system does not learn how to preheat the room. Therefore in the morning, and after lunch, the tenant remains uncomfortable until the thermal zone heats to a comfortable temperature.

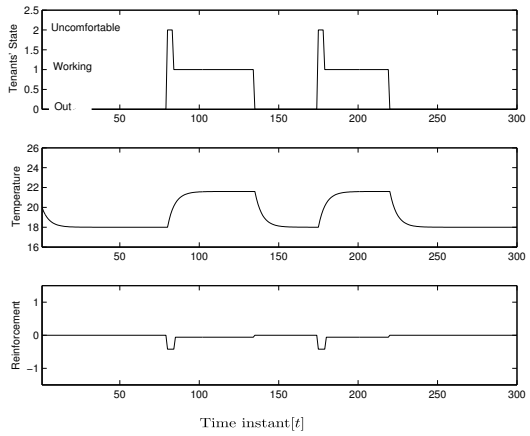

Fig. 18. Setting the temperatures using a wire table and discount factor $\gamma = 0.0$.

4.4.2. Learning how to preheat the room

If there is a delay in setting up the environmental temperature to a requested value, the BMS must learn how to request that setpoint earlier. This preheat phase can be accomplished in a certain time frame buy adjusting the discount factor γ, giving more importance to future rewards. The effects of this adjustment depend on the time resolution being used. For higher time resolutions, adjustments to γ become unnoticeable in the results. Figure 19 show the results using $\gamma = 0.9$, $w_1 = 0.01$ and $w_2 = 0.99$. The system learns how to select higher temperature earlier before the tenant arrives to avoid receiving a future negative reward. A "smoother" solution can be obtained by adding more wires, as shown in figure 20.

4.4.3. Comfort vs. Cost of heating

While using a table to hold the wires, ten experiments were conducted to evaluate the amount of time (in discrete time intervals Δ_t) the tenant is comfortable represented by $tComf$ and the total heating cost represented by thC. These functions are defined by the following set of equations:

$$tComf = \sum_{t=t_1}^{t_T} cState(t)$$

$$thC \quad = \frac{1}{T} \sum_{t=t_1}^{t_T} hC(\mathtt{T}_p(t))$$

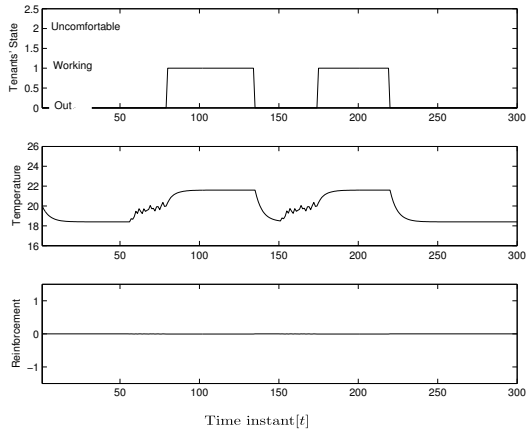

Fig. 19. Setting the temperatures using a wire table and discount factor $\gamma = 0.9$.

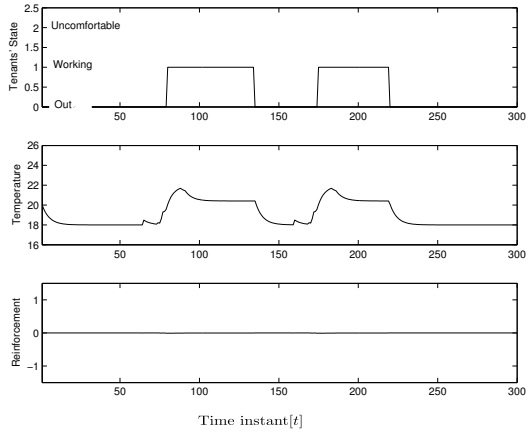

Fig. 20. Setting the temperatures using 6 wires and discount factor $\gamma = 0.9$.

with,

$$cState(t) = \begin{cases} 1, tenantState(t) = Working \\ 0, \text{Otherwise} \end{cases}$$

where $\mathtt{T}_p(t)$ is the temperature setpoint at instant t, $tenantState(t)$ is the state of the tenant and hC is given by (7).

The tenant arrives at $t = \{80, 175\}$ and leaves at $t = \{135, 220\}$ with zero variance. Table 2 shows the simulation results after a 80-day learning period, with the trade-off between the amount of time the tenant is comfortable and the total cost of heating for different pair-values $\{w_1, w_2 : w_1 \in [0, 1] \wedge w_2 = 1 - w_1\}$, $\delta = 0.99$ and $\gamma = 0.7$.

w_1	$tComf$ [Δ_t]	thC [°C Δ_t^{-1}]
0.01	100	172.33E-3
0.10	94	154.13E-3
0.20	92	157.92E-3
0.30	87	155.87E-3
0.40	90	167.26E-3
0.45	92	162.20E-3
0.50	63	132.39E-3
0.55	15	86.59E-3
0.60	0	46.14E-3
0.80	0	0.30E-3

Table 2

The number of Δ_t time intervals the tenant is comfortable ($tComf$) vs the total heating cost (thC) for different pair-values of w_1 and $w_2 = 1 - w_1$.

Table 3 shows the minimum $\Delta\mathtt{min}$, mean $\Delta\mathtt{mean}$ and maximum $\Delta\mathtt{max}$ t time instances where the tenant's discomfort state persists into the next time step i.e., the minimum, maximum and mean size of the intervals where the tenant is *Uncomfortable*.

w_1	$\Delta\mathtt{min}$ [Δ_t]	$\Delta\mathtt{mean}$ [Δ_t]	$\Delta\mathtt{max}$ [Δ_t]
0.01	0.00	0.00	0.00
0.10	3.00	3.00	3.00
0.20	4.00	4.00	4.00
0.30	1.00	1.44	3.00
0.40	1.00	2.50	4.00
0.45	1.00	2.00	3.00
0.50	1.00	6.16	14.00
0.55	2.00	12.14	35.00
0.60	45.00	50.00	55.00
0.80	15.00	50.00	55.00

Table 3

The minimum $\Delta t_{\mathtt{min}}$, maximum $\Delta t_{\mathtt{max}}$ and mean $\Delta t_{\mathtt{mean}}$ Δ_t number of time instances where the tenant is *Uncomfortable*, for different values of w_1.

5. Discussion

The results presented in section 4 show a tradeoff between comfort and cost of heating, depending on the weights set in the reinforcement function. By setting w_1 and w_2, the learning algorithm will tend to converge to a heating strategy that minimizes the penalty (negative reward) according to those weights. After a certain learning period, the BMS adjusts the zone temperature according to the tenant's preferences and occupancy patterns. This temperature can be adjusted to a

minimum setpoint where the tenant, usually, no longer complains.

The temperature of the thermal zone was not considered as a component of the state vector. This is because it is assumed that a HVAC system controller exists and can guarantee the setpoint temperature demanded by BMS.

The presented methodologies have, at least, some of the following important limitations that have to be considered:

- **Interaction with tenants**. The system learns by having the tenant interact with the thermostat. Every time the system has to learn the tenant will go through a phase where situations exist that make the tenant uncomfortable. More emphasis can be given to tenant comfort and an elaborate control scheme can override the BMS learned program, if the tenant acts on the temperature settings. The convergence rate also depends on how regular the tenants schedule is and how often he interacts with the thermostat. The presented results are for a well behaved simulated tenant that doesn't mind complaining as many times as needed for the algorithm to converge. For the sake of tenant comfort, the system should converge slowly, which means more extensive periods for continuous learning. But learned policies may become irrelevant when occupants change their behavior from time to time. This will obviously have a negative impact in terms of finding and using optimum and stable solutions, since the system explores behavioural patterns. As with humans, the better we can predict someone's behaviour, the better we can adjust to his/her habits. When those patterns change, we will have to relearn and readjust.

  Another important point is that feedback is obtained through actuation on the button interface, without knowing which tenant acted upon it. But it is very likely that most office rooms have more than one tenant with different schedules and environmental preferences. With more observability over the environment (using for example a smart video-based identification system), boundaries of comfort vs. energy optimization can be further explored considering the set of learned parameters for each specific tenant.
- **Explainability.** An intelligent system should be able to explain learned policies. One of the desired requirements for ambient intelligence is to have tenants informed about important aspects of their environment, which includes information on how energy is being used. For example, tenants can be notified about improper control of operable windows, when the HVAC is *on*, or when there are other more economical conditions that can guarantee the same comfort levels (taking advantage of e.g., solar gains and/or natural ventilation). This becomes even more important if the BMS has no control over some state variables such as doors, windows, shades, etc. A diligent tenant can follow recommendations and assume those actions, if they are properly justified by the BMS. Even with direct actuation, tenants should have some insight on why algorithms are assuming certain behaviors. In the presented solutions the policies are encoded in Q-value tables or in NN weights. Presenting an explanation to a human operator is not straightforward. To better explain this point, consider an hypothetical dialog between a human and a SB, where the human asks a building why the HVAC has been turned *off*, and the SB responds something like: "*... I was expecting that you would be attending the group meeting that is currently taking place in room 11...*". To have such a dialogue, SBs must be able to represent knowledge and reason at a symbolic level.
- **Scalability.** Using RL, with the current state representation, does not scale well and cannot cope with the complexity of a smart environment. To optimally control the operation of the HVAC, additional information has to be considered that spans out of the "room domain". For example, if the tenant has an appointment on his schedule, then the controller may decide to leave the HVAC turned *off*. To deal with such complexity, we cannot view each control problem as state-space search strategy. We have to consider a systems thinking approach [56]. Not only should we analytically partition the operation of a BMS into into smaller components, we also have to consider that everything is systemic i.e., everything interacts, affects and is affected by the things around it. The HVAC control problem cannot be solved by dealing with parts of the problem in isolation using closed and predefined state-space representations. It has to be done in concert with many other modules that interact to produce behavior.
- **Parameter tuning and convergence.** Several parameters were chosen primarily by trial and er-

ror to give the presented results. Compared to the first example, convergence was slower with the continuous state Q-learning with the NN, and was not always guaranteed. Selecting the appropriate learning rate depends on the dynamics of the environment and how is the system willing to filter out variations in behavior patterns that are not supposed to be learned. This adjustment is not always straightforward and hard to set.

6. Conclusions and Future Work

The heating, ventilation and air conditioning system is one of the the most energy-demanding systems inside a building. This paper explores how a well known reinforcement learning algorithm - Q-learning can be used to optimize the usage of this system. The solution presents the needed flexibility to adapt to the requirements of each tenant or group of tenants.

The building management system learns how to explore the limits of the temperature setpoints that tenants are able to support. A reward function is used to penalize the control agent if the heating is turned *on* or if a tenant acts upon the system, making the system learn how to keep operation parameters adjusted for energy efficiency, while keeping tenants comfortable in their working environment. Two examples were given using a discrete and continuous Q-learning approach.

The examples show how the heating system can be automatically adjusted according to the tenant preferences and occupancy patterns. A continuous state Q-learning building management system can continuously set the temperature in a way that balances maximizing tenant comfort and minimizing the energy used when the spaces are unoccupied. Simulation results were presented and limitations were discussed. The most challenging limitation include scalability issues, while extending the system to include more complex state-space representations. The system is currently not capable of explaining its decisions, but we are working on that aspect by taking into account contextual information.

Future work includes deploying the presented methodologies in a real environment, with real schedules and comfort preferences. This includes measuring temperature, setting up a tenant and HVAC interface (using, for example, a smart phone application), a BMS controller using a desktop computer and running the algorithms over a certain learning period. With a smart phone interface tenants can eventually express more than one level of discomfort by repeatedly pressing and setting/expressing their feeling as to how much should the BMS increase the current temperature (visually represented using, for example, some type of bargraph meter). Temperature control can be accomplished in a room by turning an electric heater *on* of *off* using a controllable switch.

An experimental setup is currently being developed to validate the proposed solutions and to address some of the issues discussed in the previous section in particular, the aspect of explainability and scalability. This includes setting up a network with more sensors, actuators and a restructuring of the state-space representation to include more information such as temperature information in multiple thermal-zones, state of windows and blinds, and extend the action space to operate some of those variables.

Acknowledgments

This material is based on work supported under a Portuguese National Science and Technology Foundation Strategic project [PEst-OE/EEI/LA0009/2013] and by the grant number **SFRH/BD/60481/2009**. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the National Science and Technology Foundation, or the Portuguese government.

The authors would also like recognize and thank the reviewers for their valuable contributions.

References

[1] R. Janssen, Towards energy efficient buildings in europe, Tech. rep., EuroAce - The European Alliance of Companies for Energy Efficiency in Buildings, [Online] Available: `http://euroace.org` [Accessed 18 December 2013] (June 2004).

[2] A. M. Omer, Energy, environment and sustainable development, Renewable and sustainable energy reviews 12 (2008) 2265–2300.

[3] J. Wong, H. Li, S.W.Wang, Intelligent building research: a review, Automation in Construction 14 (2005) 143–159.

[4] J. Wong, H. Li, J. Lai, Evaluating the system intelligence of the intelligent building systems part1: Development of key intelligent indicators and conceptual analytical framework, Automation in Construction 17 (2008) 284–302.

[5] J. Wong, H. Li, J. Lai, Evaluating the system intelligence of the intelligent building systems part2: Construction and validation of analytical models, Automation in Construction 17 (2008) 303–321.

[6] A. Aztiria, A. Izaguirre, J. C. Augusto, Learning patterns in ambient intelligence environments: a survey, Artificial Intelligence Review 34 (1) (2010) 35–51.

[7] M. M. Levine, Energy efficiency improvement utilizing high technology: An assessment of energy use in industry and buildings; report and case studies, London SW1A 1HD, United Kingdom: Technical report, World Energy Council, 34 St James's Street.

[8] C. J. Watkins, P. Dayan, Technical note: Q-learning, Machine Learning 8 (3-4) (1992) 279–292.

[9] R. S. Sutton, A. G. Barto, Introduction to Reinforcement Learning, 1st Edition, MIT Press, Cambridge, MA, USA, 1998.

[10] A. Barto, P. Anandan, Pattern-recognizing stochastic learning automata, IEEE Transactions on Systems, Man and Cybernetics SMC-15 (3) (1985) 360–375.

[11] D. H. Ackley, Advances in neural information processing systems 1, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 1989, Ch. Associative learning via inhibitory search, pp. 20–28.

[12] R. Allen, Developing agent models with a neural reinforcement technique, in: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Vol. 1, 1989, pp. 206–207.

[13] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach (2nd Edition), Prentice Hall, ISBN-10: 0137903952, 2002.

[14] L. C. B. III, A. H. Klopf, Reinforcement learning with high-dimensional, continuous actions, Tech. rep., Wright Laboratory (1993).

[15] A. Dounis, C. Caraiscos, Advanced control systems engineering for energy and comfort management in a building environment - a review, Renewable and Sustainable Energy Reviews 13 (6–7) (2009) 1246–1261.

[16] D. Naidu, C. G. Rieger, Advanced control strategies for HVAC&R systems - an overview: Part 2: Soft and fusion control, HVAC&R Research 17 (2) (2011) 144–158.

[17] R. Alcalá, J. M. Benítez, J. Casillas, O. Cordón, R. Pérez, Fuzzy control of HVAC systems optimized by genetic algorithms, Applied Intelligence 18 (2) (2003) 155–177.

[18] R. Alcalá, J. Casillas, O. Cordón, A. González, F. Herrera, A genetic rule weighting and selection process for fuzzy control of heating, ventilating and air conditioning systems, Engineering Applications of Artificial Intelligence 18 (3) (2005) 279–296.

[19] V. Congradac, B. Milosavljevic, J. Velickovic, B. Prebiracevic, Control of the lighting system using a genetic algorithm, Thermal Science 16 (suppl. 1) (2012) 237–250.

[20] K. Fong, V. Hanby, T. Chow, System optimization for HVAC energy management using the robust evolutionary algorithm, Applied Thermal Engineering 29 (11-12) (2009) 2327–2334.

[21] J. Teeter, M.-Y. Chow, Application of functional link neural network to HVAC thermal dynamic system identification, IEEE Transactions on Industrial Electronics 45 (1) (1998) 170–176.

[22] C. Hernandez S, R. Romero, D. Giral, Optimization of the use of residential lighting with neural network, in: (CISE), 2010 International Conference on Computational Intelligence and Software Engineering, 2010, pp. 1–5.

[23] E. Sierra, A. Hossian, D. Rodríguez, M. García-Martínez, P. Britos, R. García-Martínez, Optimizing building's environments performance using intelligent systems, in: Proceedings of the 21st international conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems: New Frontiers in Applied Artificial Intelligence, 2008, pp. 486–491.

[24] Y. Penya, C. Borges, D. Agote, I. Fernandez, Short-term load forecasting in air-conditioned non-residential buildings, in: (ISIE), 2011 IEEE International Symposium on Industrial Electronics, 2011, pp. 1359–1364.

[25] P. Angelov, A fuzzy approach to building thermal systems optimization, in: 8th International Fuzzy Systems Association World Congress, 1999.

[26] H.-B. Kuntze, T. Bernard, A new fuzzy-based supervisory control concept for the demand-responsive optimization of HVAC control systems, in: Proceedings of the 37th IEEE Conference on Decision and Control, 1998., Vol. 4, 1998, pp. 4258–4263.

[27] S. Soyguder, H. Alli, An expert system for the humidity and temperature control in HVAC systems using ANFIS and optimization with fuzzy modeling approach, Energy and Buildings 41 (8) (2009) 814–822.

[28] V. Singhvi, A. Krause, C. Guestrin, J. H. Garrett, Jr., H. S. Matthews, Intelligent light control using sensor networks, in: Proceedings of the 3rd international conference on Embedded networked sensor systems, 2005, pp. 218–229.

[29] E. Sierra, A. Hossian, D. Rodríguez, M. García-Martínez, P. Britos, R. García-Martínez, Fuzzy control for improving energy management within indoor building environments, in: Electronics, Robotics and Automotive Mechanics Conference. CERMA, 2007, pp. 412–416.

[30] P. Albertos, A. Sala, Fuzzy logic controllers. advantages and drawbacks, in: XIII Congreso de la Asociación Chilena de Controlo Automático, Vol. 3, 1998, pp. 833–844.

[31] J. Godjevac, Comparative study of fuzzy control, neural network control and neuro-fuzzy control, Tech. rep., École Polytechnique Fédérale de Lausanne (February 1995).

[32] L. C. Baird, M. E.Harmon, A. H. Klopf, Reinforcement learning: An alternative approach to machine intelligence, Tech. rep., Wright Laboratory (1996).

[33] L. P. Kaelbling, M. L. Littman, A. W. Moore, Reinforcement learning: a survey, Journal of Artificial Intelligence Research 4 (1) (1996) 237–285.

[34] M. Kintner-Meyer, A. F. Emery, Optimal control of an HVAC system using cold storage and building thermal capacitance, Energy and Buildings 23 (1) (1995) 19–31.

[35] K. Dalamagkidis, D. Kolokotsa, Reinforcement learning for building environmental control, in: C. Weber, M. Elshaw, N. M. Mayer (Eds.), Reinforcement Learning - Theory and Applications, I-Tech Publications, 2008, Ch. 15, pp. 283–294.

[36] International Organization for Standardization, Ergonomics of the thermal environment - Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort criteria, EVS-EN ISO 7730:2006.

[37] P. O. Fanger, Thermal Comfort: Analysis and applications in environmental engineering, McGraw-Hill, New York, 1972.

[38] J. Van Hoof, Forty years of fanger's model of thermal comfort: comfort for all?, Indoor Air 18 (3) (2008) 182–201.

[39] L. Schellen, M. Loomans, W. van Marken Lichtenbelt, A. Frijns, M. de Wit, Assessment of thermal comfort in relation to applied low exergy systems, in: Adapting to Change: New Thinking on Comfort, 2010.

[40] P. A. Ruiz, J. G. Martín, L. O. Jesús M. Sanz, New model for the search for comfort through surveys, in: WSEAS Transac-

tions on Circuits and systems, Vol. 11, 2012, pp. 125–135.

[41] M. A. Humphreys, Standards for Thermal Comfort, Chapman & Hall, 1995, Ch. Thermal comfort temperatures and the habits of Hobbits, pp. 3–13.

[42] E. Mathews, D. Arndt, C. Piani, E. van Heerden, Developing cost efficient control strategies to ensure optimal energy use and sufficient indoor comfort, Applied Energy 66 (2) (2000) 135–159.

[43] T. Williamson, P. Riordan, Thermostat strategies for discretionary heating and cooling of dwellings in temperate climates, in: 5th IBPSA Building Simulation Conference, 1997, pp. 1–8.

[44] U. Rutishauser, J. Joller, R. Douglas, Control and learning of ambience by an intelligent building, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 35 (1) (2005) 121–132.

[45] T. Peffer, M. Pritoni, A. Meier, C. Aragon, D. Perry, How people use thermostats in homes: A review, Building and Environment 46 (2011) 2529–2541.

[46] A. Meier, Thermostat interface and usability: A survey, Tech. rep., Lawrence Berkeley National Laboratory, LBNL Paper LBNL-4182E (2011).

[47] T. Peffer, M. Pritoni, A. Meier, C. Aragon, D. Perry, How people use thermostats in homes: A review, Building and Environment 46 (12) (2011) 2529–2541.

[48] S. Wang, X. Xu, Optimal and robust control of outdoor ventilation airflow rate for improving energy efficiency and IAQ, Building and Environment 39 (7) (2004) 763–773.

[49] P. Boait, R. Rylatt, A method for fully automatic operation of domestic heating, Energy and Buildings 42 (1) (2010) 11–16, international Conference on Building Energy and Environment (COBEE 2008).

[50] M. Gupta, S. S. Intille, K. Larson, Adding gps-control to traditional thermostats: An exploration of potential energy savings and design challenges, in: Proceedings of the 7th International Conference on Pervasive Computing, Pervasive '09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 95–114.

[51] R. S. Sutton, Learning to predict by the methods of temporal differences, Machine Learning 3 (1988) 9–44.

[52] M. Coggan, Exploration and exploitation in reinforcement learning, in: Fourth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'01), Yokosuka City, Japan, 2001.

[53] C. Gaskett, D. Wettergreen, A. Zelinsky, Q-learning in continuous state and action spaces, in: Australian Joint Conference on Artificial Intelligence, Springer-Verlag, 1999, pp. 417–428.

[54] H. van Hasselt, M. A. Wiering, Reinforcement learning in continuous action spaces, in: Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning, 2007, pp. 272–279.

[55] A. I. Dounis, C. Caraiscos, Fuzzy comfort and its use in the design of an intelligent coordinator of fuzzy controller-agents for environmental conditions control in buildings, Uncertain Systems 2 (2) (2008) 101–112.

[56] G. Bartlett, Systemic thinking a simple thinking technique for gaining systemic focus, in: The International Conference on Thinking - Breakthroughs 2001, 2001.