A multi-agent reinforcement learning approach to energy and comfort management

Mustafa Shaikh

under the supervision of

Dr. Chi-Guhn Lee, Mechanical & Industrial Engineering, University of Toronto

April 2017

A multi-agent reinforcement learning approach to energy and comfort management

Mustafa Shaikh

under the supervision of

Dr. Chi-Guhn Lee, Mechanical & Industrial Engineering, University of Toronto

April 2017

Abstract

This paper presents a study in which a retail store aims to reduce the energy cost of its heating, ventilating and air conditioning (HVAC) systems, while providing a comfortable shopping environment to its customers. Various models which describe the underlying physical systems and processes are employed in combination with an optimization problem in order to determine the optimal policy for the store's HVAC system operation. The optimization problem was formulated as a multi-agent Markov decision process, in which each agent controls its own HVAC such that the system-wide cost can be optimized. The solution is to provide a complete policy by which the store can determine the temperature set-point of the three HVAC systems in response to observing various system states involving the external and internal temperatures, and the estimated occupancy level. The complexity of the problem is high and therefore a variant of a well known reinforcement learning algorithm is used to make effective use of available historical data and solve the optimization task. To reduce the complexity further, we devised a threshold-based state space reduction method by partitioning the temperature, humidity and population range into intervals. We present numerical studies and visualizations to demonstrate the energy savings and increased comfort levels that can be achieved by the optimal control policy found by the proposed algorithm.

Acknowledgments

I would like to thank Simone Stancari and Energy Way srl. for presenting the research project that is the subject of this thesis, and for the work that they contributed to it.

I would also like to thank Dr. J.S. Kim and Babatunde Giwa for the work they contributed to this project.

Special thanks to Professor Chi-Guhn Lee, for providing me with the opportunity to partake in this research project, and for his invaluable guidance and assistance throughout the entire duration of the project.

Table of Contents

Table of Figures

1 Introduction

Energy efficiency has, over the last few decades, increasingly become a topic of research and discussion, as we attempt to reduce our carbon emissions and cut energy costs. With the rise of the 'smart building', numerous sensors monitor and optimize energy usage based on parameters such as temperature, humidity, occupancy trends and more (Intel, 2016). However, buildings or other enclosed spaces that cannot implement a complete packaged optimization system (Shift Energy, 2016), and that wish to improve their energy efficiency, also need a way of achieving this at low cost. It is to address this issue that this study is being carried out.

For the purposes of this thesis, a particular retail store* in Modena, Italy, will be used as the environment for the study. The research uses a large amount of actual data obtained from the industry partner. The ultimate aim is to minimize an objective function, which consists of the weighted sum of energy cost - obtained through electricity usage data - and a penalty for customer discomfort. The approach uses dynamic optimization under uncertainty, and so over time, delivers a global minimum cost solution.

This will be achieved by first creating models according to which optimization decisions will be made. These models include cost models associated with changes in the HVAC system and determination of a measure for comfort (more detail to follow). Using the information extracted from these models, the agents - in this case the supermarket's 3 HVAC system controllers - will make decisions based on a decentralized Markov decision process (Beynier, 2010) to optimize the operation of the HVAC system, which only controls temperature. A reinforcement learning algorithm will be implemented to achieve this.

Much research has already been conducted in the field of energy use optimization, and this study makes use of models developed in the literature. However, there are some notable differences between existing research and this study, mostly due to the fact that the work being undertaken is for a specific application. For example, some of the existing research has been focused on optimizing the ventilation rate in the enclosed environment (Leephakpreeda, 2001) (Wang et al., 1999), whereas the research being carried out aims to optimize only the setpoint temperature.

*Herein referred to as 'retail store'; further details subject to confidentiality agreement

2 Formulation and Models

Research available in the existing literature has been used to develop relevant models pertaining to the various aspects of the study. These can be categorized as follows:

1. The underlying mathematical framework used for the Markov Decision Process (MDP) and reinforcement learning algorithm
2. Estimation of occupancy using indoor carbon dioxide (CO_2) measurements
3. Modelling of human thermal comfort

This section presents a brief description of the physical system in the retail store, which will provide additional context for the review of the literature that follows.

2.1 System description

The system configuration of the system is shown in Figure 1. A simplified overview of the refrigeration cycle is as follows: two chillers (chillers 1, 2) produce cold water which is distributed to the three connected HVACs. The cold water undergoes heat exchange with the air, which is then circulated to the individual regions (yellow, orange, and green) by the ventilation systems to reach the desired internal temperature setpoint.

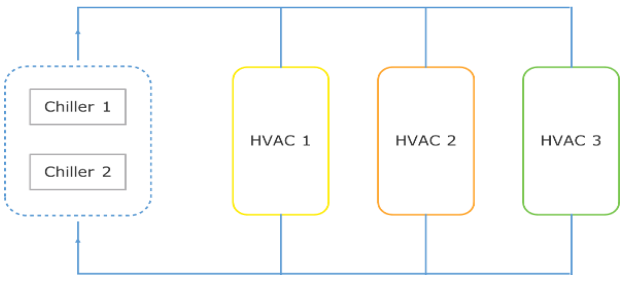


Figure 1: Cooling mechanism in retail store

The physical layout of the HVAC system for the retail store is shown in Figure 2. Currently, the individual HVAC systems are controlled identically in terms of temperature setpoints regardless of the different thermal and occupancy conditions of the respective areas. This is a sub-optimal

policy as the potential savings associated with increasing temperature setpoint based on occupancy and thermal conditions are overlooked.

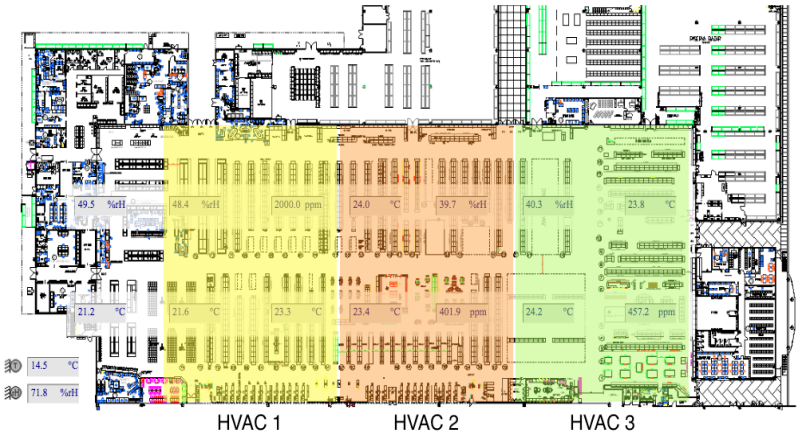


Figure 2: Physical layout of the HVAC systems

This gives rise to a need for adaptive control of the temperature setpoints to allow for effective energy utilization while maintaining the desired comfort levels, during the summer period from 9am to 10pm daily. Note that only the cooling system will be considered in this study as the heating system is switched off during the summer months.

2.2 Mathematical framework

2.2.1 Markov Decision Processes (MDP)

A Markov Decision Process (herein referred to as MDP) is a class of sequential decision making models under uncertainty, which allows agents to 'decide how to act in a stochastic environment in order to maximize a given performance measure' (Garcia & Rachelson, 2010). There are many classes of MDP's; the ones used in this research will be a single agent MDP and a multi-agent decentralized MDP. In this case, each of the supermarket's 3 HVAC systems correspond to an agent – the ultimate goal of this research is to determine a policy which dictates each HVAC system's temperature set point at every time step.

MDP's are defined as 'controlled stochastic processes that satisfy the Markov property' (Garcia & Rachelson, 2010) – that is, the system can be fully described by defining possible system

states, actions, rewards for taking those actions, and state transition probabilities. This can be represented concisely by the following tuple $\{S, A, T, p, r\}$:

S – *state space in which the system or process occurs*
A – *set of all possible actions that can be taken in any given state*
T – *set of time steps where decisions are to be made*
p – *probability of transitioning from a given state to another*
R – *reward for taking an action a in state s*

There can be finite and infinite time horizons; in this case, since the process is study is conducted for the summer months, the time horizon will last approximately 3 months. Furthermore, since this is a case of dynamic optimization, that is, an optimal policy is to be found which translates any state to an optimal action, there is no 'goal state' which is to be reached at the end of the time horizon. A simple flow diagram illustrating an MDP is shown in below. State i transitions to state $i+1$ with probability $P(s_{i+1}|s_i, a)$ when action a is taken – reward $r(s, a)$ is earned immediately and there is a unit increase in time.

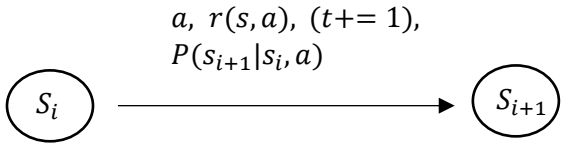


Figure 3: Basic MDP illustration

Since the system is formulated as an MDP, which satisfies the Markov property, the dynamics of the system can be described using transition probabilities as mentioned above. The transition probabilities associated with the various states in the system can be compactly represented in transition matrices. These matrices specify the probability that the system moves from a given state to any other state, and can be used to predict the evolution of the system as a response to a particular action.

2.2.2 Reinforcement learning

Once the problem has been formulated as an MDP, an optimal policy using historical system data can be found using a reinforcement learning algorithm, specifically Q-learning, presented by Watkins (Watkins, Learning from delayed rewards, 1989). An optimal policy is one which informs the agent of the best action to take in any given state, so as to arrive at a global optimal solution. At the core of the Q-learning algorithm is a value iteration which seeks to find the total expected future reward for taking an action in a given state. The importance given to future rewards is reflected in a weight, called the discount factor $\gamma \epsilon [0,1]$, where 0 assigns no importance and 1 assigns maximum importance to future rewards. The Q-learning algorithm is as follows (Watkins, Learning from delayed rewards, 1989):

$$Q_n(s,a) = (1-\alpha_n)\, Q_{n-1}(s,a) + \alpha_n[R(s,a) + \gamma * max_{a'} Q_{n-1}(s',a')]$$

$Q_n\,(s,a)$– $Q-value\ at\ training\ episode\ n\ associated\ with\ taking\ an\ action\ a\ in\ state\ s$
$Q_n\,(s',a')$ – $Q-value\ associated\ with\ taking\ an\ action\ a'\ in\ the\ next\ state\ s'$
α_n – $learning\ rate\ at\ training\ episode\ n$
$R(s,a)$ – $immediate\ reward\ for\ taking\ action\ a\ in\ state\ s$
γ – $discount\ factor$

The Q-learning algorithm is guaranteed to converge with probability 1 as $n \rightarrow \infty$, subject to the following conditions:

1. Rewards must be bounded i.e. $|R_n| \leq R \ \ \forall\, n$
2. Learning rate α_n subject to $0 \leq \alpha_n < 1,\ \sum_{i=1}^{\infty} \alpha_n^i = \infty,\ \sum_{i=1}^{\infty} (\alpha_n^i)^2 < \infty$

A function such as $\alpha = \frac{1}{k}$ satisfies the above conditions. The full proof of the convergence is presented in a technical note co-authored by Watkins. The interested reader is encouraged to read further (Watkins & Dayan, Technical note: Q-Learning, 1992).

An important part of the learning process is the *exploration strategy* - in other words, the method by which the agent explores the environment to gain experience. This is important as the agent

must have sufficient experience in its environment to be able to determine an optimal policy for the various possible states it may find itself in. This topic will be discussed in more detail in section 3.6 relating to the implementation of the Q-learning algorithm.

The Q-learning algorithm is inherently model free; the agent does not require information about the plant to determine the optimal policy (Watkins, Learning from delayed rewards, 1989). In the context of this research work, historical plant data is available, and is not constantly being updated at this point. Therefore, it is useful to use that plant information in the form of transition probability matrices, as introduced above. The Q-learning algorithm requires information on the next state in order to find the maximum Q-value across all actions in the next state (see last term of the algorithm). To that end, the transition matrices can be used to probabilistically determine the next state during training.

2.3 CO_2 based occupancy estimation

Estimating the occupancy of the supermarket is a key component of the research work. Since the aim is to optimize the energy cost whilst maintaining customer comfort, it is important to estimate the number of people in the supermarket. The penalty for customer discomfort will be proportional to the number of people affected; fewer people present, say at off peak times, means that the penalty for causing some discomfort to those few people is lower than causing discomfort to people at peak times.

As outlined earlier, many models in the literature make use of CO_2 measurements to inform ventilation policies. Some assume sensor rich environments (Yang, 2013), and others propose the installation of new sensing systems. However, these do not suit the purposes of this research. Models which predict occupancy solely using CO_2 measurements are required. Ito and Nishi (2012) present a set of equations derived from conducting a mass balance on an enclosed space. The equation is as follows (Ito & Nishi, 2012):

$$n = \frac{Q}{k\left(1 - e^{-\frac{Q}{V}(i-s)}\right)}(C_i - C_0 - (C_s - C_0)e^{-\frac{Q}{V}(i-s)})$$

$n - number\ of\ people$

$Q\ (m^3/h) - ventilation\ rate$

$V\ (m^3) - volume\ of\ the\ enclosed\ space$

$i, s\ (s) - initial\ time, current\ time$

$k\ (m^3/h.person) - CO_2\ emissions\ per\ person$

$C_s - CO_2\ concentration\ at\ time\ s$

$C_0 - CO_2\ concentration\ in\ empty\ room$

The value for Q was assumed to be 0.06 cfm/ft^2 according to ASHRAE guidelines (ASHRAE, 2003), and k was assumed to be 0.008 L/s (Emmerich & Persily, 2001). Although the authors present a new sensing system as well, these equations will be used with the existing sensory network to provide occupancy estimation.

2.4 Modelling of human thermal comfort

Modelling of human comfort is an important part of the research work. The comfort of customers is of paramount importance to the supermarket's management, and represents one of two variables that are part of the objective function, the other being energy cost. As outlined above, there is a cost associated with causing discomfort to customers, and so it is instructive to model and quantify human thermal comfort. Using information on how comfortable people will be in certain situations will help develop an accurate objective function.

Arguably the most influential research on indoor human thermal comfort was conducted by P.O. Fanger. His seminal publication, 'Thermal Comfort: Analysis and Applications in Environmental Engineering' (Fanger, 1970), has informed subsequent work in academia as well as industry. The American Society of Heating, Refrigerating and Air Conditioning Engineering (herein referred to as ASHRAE) has incorporated guidelines based on Fanger's research into its thermal environmental conditions code. The section that is relevant to this research work introduces the concept of the 'Predicted Mean Vote (PMV)' and 'Predicted Population Dissatisfied (PPD)'. The PMV represents a comfort level on an integer scale from -3 (very cold), to +3 (very hot), where 0

represents thermal neutrality, and is the ideal value. The equation for PMV is shown below (Fanger, 1970). It consists of many variables – the definitions of the more important ones have been shown for brevity.

$$PMV = [0.303e^{-0.036M} + 0.028]\{(M - W) - 3.96 * 10^{-8} f_{cl}[(t_{cl} + 273)^4 - (t_r + 273)^4] - f_{cl} h_c (t_{cl} - t_a) - 3.05[5.73 - 0.007(M - W) - p_a] - 0.42[(M - W) - 58.15] - 0.0173M(5.87 - p_a) - 0.0014M(34 - t_a)\}$$

$where$

$$f_{cl} = \begin{cases} 1.0 + 0.2I_{cl} \\ 1.05 + 0.1I_{cl} \end{cases}$$

$$t_{cl} = 35.7 - 0.0275(M - W) - R_{cl}\{(M - W) - 3.05[5.73 - 0.007(M - W) - p_a] - 0.42[(M - W) - 58.15] - 0.0173M(5.87 - p_a) - 0.0014M(34 - t_a)\}$$

$$R_{cl} = 0.155I_{cl}$$

$$h_c = 12.1(V)^{1/2}$$

f_{cl} - $clothing\ factor$

I_{cl}- $clothing\ insulation\ (clo)$

M - $metabolic\ rate\ (w/m^2)$

T_a - $air\ temperature\ (^oC)$

ASHRAE recommends that the PMV value in an indoor environment should be kept within ±0.5 (American Society of Heating, Refrigeration and Air Conditioning Engineers, 2013). The PPD value estimates the percentage of population that will be dissatisfied with the conditions, although this parameter will not be used.

3 Implementation

This section discusses implementation methodology, including challenges that were encountered, and how they were overcome. All work to date has been carried out on a single HVAC system (HVAC 2) and will yield the optimal policy for HVAC 2, but the ultimate goal is to determine a

global optimal policy across all 3 systems. The implementation is extendable to each of the three individual HVAC systems.

3.1 Processing the data

Sensor data for 3 months during the summer of 2015 and 2014 was provided by the industry partner, mainly monitoring the following parameters for all 3 supermarket sections, in 30-minute time intervals; internal/external temperature, internal/external relative humidity, internal CO_2 concentration and internal setpoint temperature. However, the models required to obtain an optimal solution made use of all of the above data, and with 10,000 data points for each parameter, defining the states of the system for the MDP framework would lead to an enormous state space. This in turn would lead to prohibitively high complexity when seeking the optimal solution, so the data was discretized into a finite number of bands to reduce the size of the state space.

After initial examination of the data, it was decided that there were 3 important parameters to include in the state definition; internal temperature, external temperature and occupancy of the supermarket section. Although other parameters were used in the various models, these were omitted from the state space – this will be discussed further in later sections. The raw data was first passed into Python, where the rest of the operations were performed. The internal temperature was split into 5 intervals, with integer values from 22^oC to 26^oC; this covered the vast majority of the range of internal temperatures in the data. Any values below 22^oC or above 26^oC were clipped to 22^oC and 26^oC respectively. External temperature was discretized into 3 numbered intervals, with each interval represented by a single value at the midpoint of the interval; [1: 20ºC, 2: 24ºC, 3: 28ºC]. The occupancy level was similarly split into 5 numbered intervals; [1: 4 people, 2: 10, 3: 15, 4: 22 and 5: 30]. This discretization resulted in a total of 75 possible system states, greatly reducing the size of the original state space.

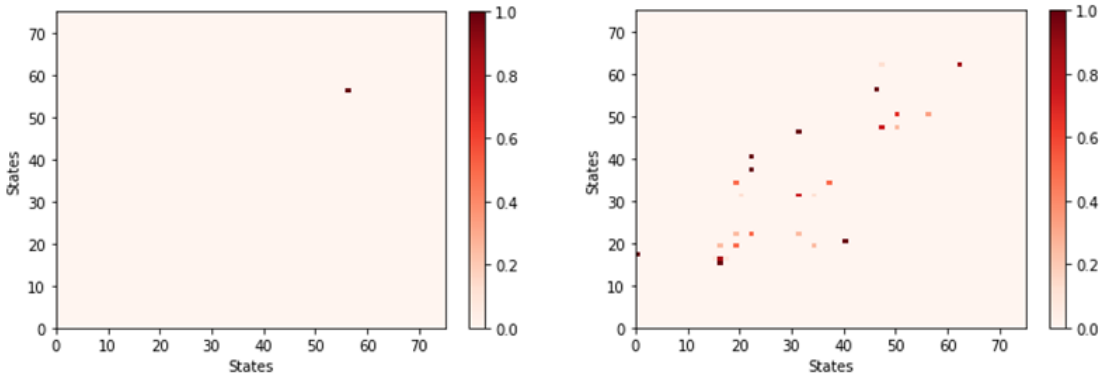
In order to perform operations on the data more easily, the state information was represented by a single decimal number in the form $x.yz$, where $x \in [22, 23, 24, 25, 26]$ $y \in [1, 2, 3, 4, 5]$ $z \in [1, 2, 3]$ correspond to each internal temperature, occupancy level and external

temperature interval respectively. For example, 22.11 represents state 1, with [$T_{int} = 22^oC$, $n = 4\ people$, $T_{ext} = 20^oC$].

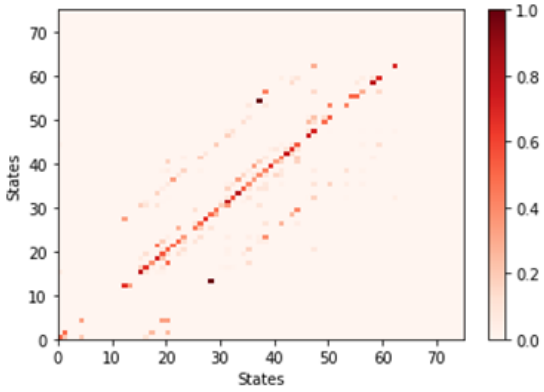
3.2 Transition probability matrices

As outlined previously, transition probability matrices are a fundamental part of an MDP formulation. Once the sensor data was processed, the transition probability matrices were determined for each setpoint temperature [22ºC – 26ºC] (since there are 75 total states, the transition matrices all have dimension 75x75). The process is as follows: the data is read in, discretized, and stored in a [n x 1] vector, where n is the number of data points. A list of all possible states, in order from 22.11 to 26.53, is generated and stored in a 75 x 1 vector. A dictionary containing key value pairs of the raw state number and encoded state number is used to encode the raw state data from state 1 to 75. For example, if the input is 22.12, the output will be 2, as 22.12 corresponds to the 2nd out of 75 possible states. The number of transitions between each state is then counted, stored in the transition matrix and each value divided by the sum of its row to normalize it. This results in the transition probability matrix, which will be used to probabilistically determine the next state of the system in the training process of the learning algorithm to be detailed later. Figure 4 shows a visualization of the nature of select transition matrices. Coloured blocks represent non zero values in the transition matrix – note the strong correlation between similar states. The non-zero values occur almost exclusively along a single axis, with low variance.

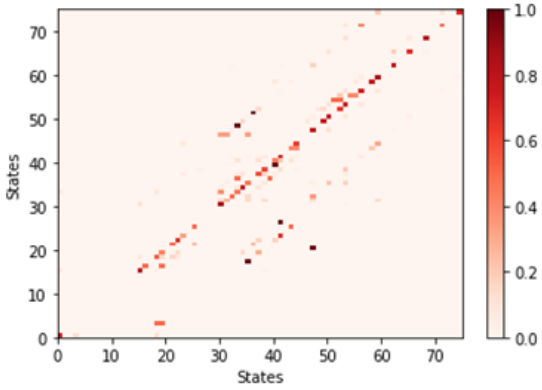
Setpoint = 22^oC Setpoint = 23^oC



Setpoint = 24^oC



Setpoint = 25^oC



Setpoint = 26^oC

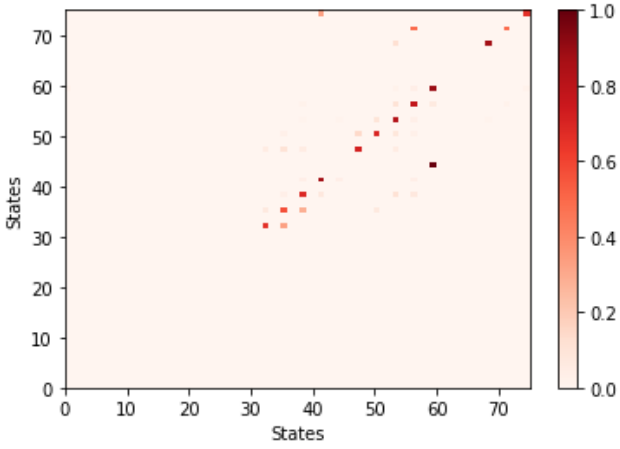


Figure 4: Visualizations of transition probability matrices

A more intuitive visualization can be seen in Figure 5 – it shows the probabilities with which the system moves from a certain internal temperature to any other internal temperature, for a setpoint temperature of 24^oC. From Figure 5, we see that the regions of highest probability are as expected given a setpoint temperature of 24^oC. For example, there is a high probability associated with transitioning from an internal temperature of 24^oC at time t, to 24^oC at time $t+1$, given a setpoint of 24^oC. We also see that internal temperatures at time t of 22^oC and 26^oC show relatively higher probabilities of moving closer to 24^oC at time $t+1$. Although we may

expect there to be an even higher probability of moving to the setpoint within a time step, it is plausible that the measured ambient temperature would exhibit a lag of more than one time step from when the setpoint was changed. The visualizations for transition matrices resulting from the other setpoints are very similar.

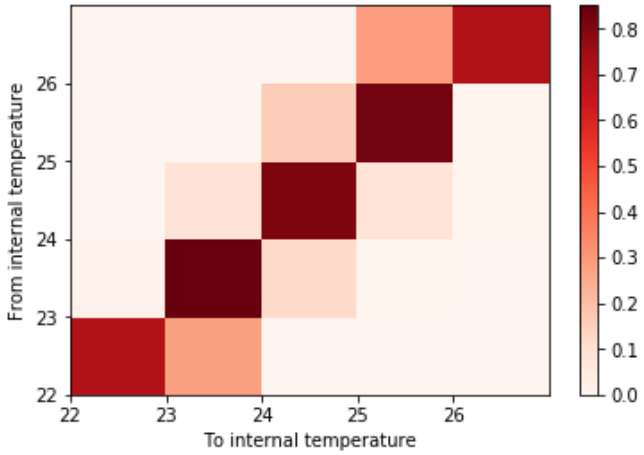


Figure 5: Alternative visualization of transition probability matrix

3.3 Thermal comfort model

Given the values of the input variables to the comfort model introduced earlier, it is possible to compute a value of thermal comfort in any given environmental condition. Standard values for some of the variables were used, for example, 2 met is the metabolic rate of the average person whilst walking (Emmerich & Persily, 2001), and so this value was used in the model. For variables that have a greater impact on the output of the model but are not part of the state information, such as internal relative humidity, an alternative needed to be found. To address this issue, a sensitivity analysis was performed on the model, with the variation in PMV value measured for several values of relative humidity. It was found that from a change in relative humidity from 30% to 60%, the PMV value was only affected by ±0.4. However, such a large change in internal relative humidity is rare; the lowest value recorded in the data is 35.5%, but the mean value is 54.5%, with a standard deviation of 6.9%. Due to the relatively low standard deviation of humidity values, the internal relative humidity was approximated to remain constant at the mean value of 54.5%, with the assumption that changes in the humidity on the scale of the standard deviation would cause negligible changes in the PMV. Using similar approximations

and standard values, the PMV can be calculated at each time step, and plays an integral role in the objective function detailed later. Figure 6 shows the dynamics of the thermal comfort model.

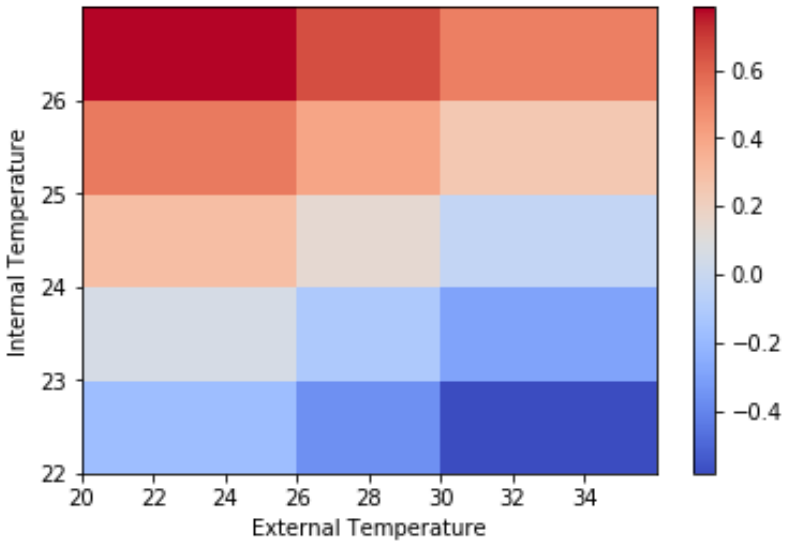


Figure 6: Visualization of the thermal comfort model

Maximum comfort is shown by pale colours (corresponding to regions immediately surrounding 0.0 on the colour scale), feeling cold is shown by shades of blue (darker blue is colder), and feeling hot is shown by shades of red (darker red is hotter). The model behaves as expected; when the external temperature is high, people will on average wear less clothing, and so will feel cold in low internal temperatures, hence the dark blue patch at ($T_{ext} = 32^oC, T_{int} = 22^oC$). Conversely, at low external temperatures, people will on average wear more clothing, and so will feel hot in high internal temperatures, hence the dark red patch ($T_{ext} = 20^oC, T_{int} = 26^oC$). The ideal region is where the image shows pale coloured squares – these represent areas of maximum comfort. Note that this occurs when the difference between internal and external temperatures is low.

3.4 CO_2 based occupancy estimation

Using the set of equations introduced earlier, the occupancy level of each region of the store can theoretically be determined. Upon implementation, however, it was found that the actual values output by the equations were inaccurate by several orders of magnitude. This is most likely

because the data available from the sensors was not completely equivalent to the variables defined in the equation in the literature. However, further investigation revealed that the pattern of values was similar to the predicted pattern of occupancy; a periodic function with variable peaks and troughs, which reflect the variation in occupancy at various times of day. To make use of the model, the output values were scaled to a range of values with a mean of 12 people. Note also that some values in the output are negative – all such values were clipped to 0. The output also had undesirable noise which further distorted the values; this was driven by noise in the original CO_2 data. To that end, a Savitsky-Golay filter (Savitsky & Golay, 1964) was applied to remove these components. The filter works to smooth the data by fitting a specified n^{th} order polynomial to successive windows of the data set – the size of this window is also specified by the user. For a more intuitive understanding, the method can be thought of as taking a moving average of the data with specified window size.

The smoothing procedure was carried out with varying degrees of polynomials, and the output which appeared most similar to the original output was selected. In this case, it was an 8^{th} order polynomial with a window size of 101. Figure 7 shows a sample output with a polynomial of degree 3, whilst Figure 8 shows the output which was eventually used.

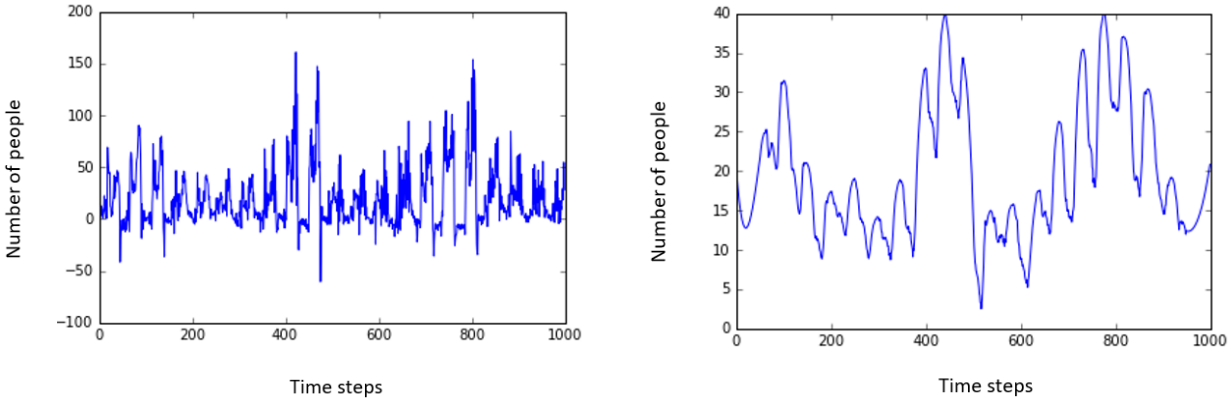


Figure 7: Unfiltered data vs. filtered data - 3rd order polynomial with window size of 101

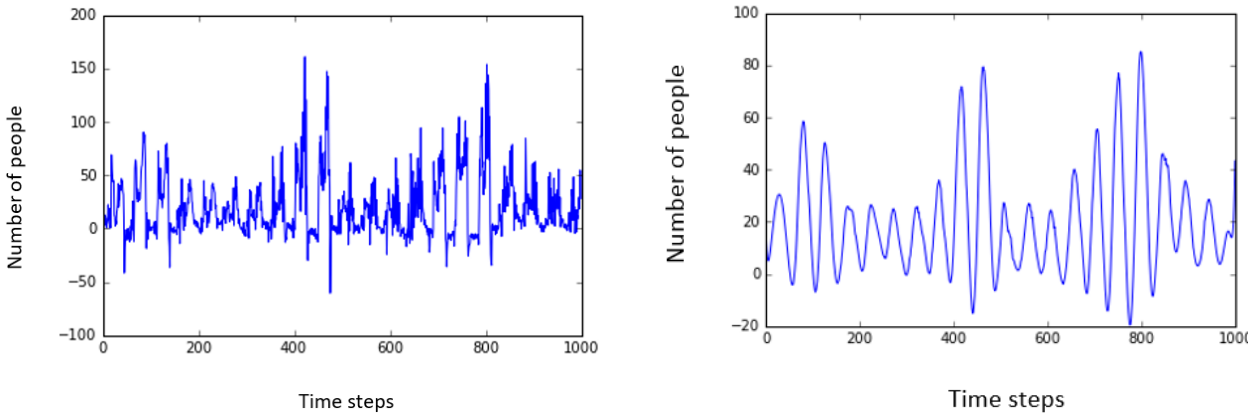


Figure 8: Unfiltered data vs. filtered data - 8th order polynomial with window size of 101

As can be seen from Figures 7 and 8, the filter adequately smoothed the output.

3.5 Cost models

An important component of the Q-learning algorithm used to determine the optimal policy, is the objective, or cost function, which is to be minimized over time. The cost function consists of two terms – an energy cost term, and a discomfort penalty. To simplify the dynamics of the learning algorithm, both terms have been combined into a single cost function, despite having different units.

3.5.1 Discomfort penalty

In order to combine the two costs in a meaningful manner, the discomfort penalty must be appropriately scaled. To achieve this, two weights are applied to the comfort value, which essentially convert customer discomfort to a dollar value and scale it according to the occupancy level, thereby quantifying the cost of discomfort. A policy which causes discomfort to more people will therefore be penalized more strongly than one which causes discomfort to fewer people, and more importantly, the effect of discomfort on the optimal policy will be equal to the effect of the energy cost.

The first weight is simply equal to the number of people in the given region at any time. The second weight scales the discomfort penalty so that it has a similar effect on the optimal policy to

the energy cost, despite having values on a lower order of magnitude. This is achieved by variance equalization – the variance of the energy cost function computed across all states was compared to the variance of the discomfort penalty before it was weighted by the occupancy level, for several values of the discomfort penalty weight. The weight that was selected was one which caused the variance of both the energy cost function and discomfort penalty to be equal.

To gain an intuition for the method of variance equalization for selecting the weight, and to understand why it is a more appropriate method than an approach such as mean equalization, consider the following scenario. Suppose there are two sets of data, with different distributions. The first set is described by the following parameters: $\mu = 100, \sigma^2 = 30$, and the second set is described by $\mu = 50, \sigma^2 = 1$. Using the approach of mean equalization, the mean of the second set would be scaled to $\mu = 100$ by doubling each data point, and the variance would increase to $\sigma^2 = 2$. Now suppose a random sample is taken from both distributions and added, and the process repeated several times. The question arises, which distribution generates samples that have a greater effect on the final outcome of repeated additions? The answer lies in the variation of each sample. We see that the distribution of the second data set has a much lower variance than that of the first distribution – therefore, when a random sample is taken from each distribution and added to each other, we expect that the sample drawn from the second distribution will have a value very close to 100, and so we can infer the value of the sample taken from the first distribution by simply subtracting 100 (expected value of the sample drawn from the second distribution) from the result. Say the outcome of one such addition is 160; from the mean and variance of each distribution, we see that it is likely that this value was caused by a value of 60 sampled from the first distribution, and a value of around 100 sampled from the second distribution. Over n addition episodes, the final value will essentially be a function of the first distribution, as the variations in samples drawn from the second distribution will be effectively ignored as the scale of variation of those samples is much smaller than those of the first distribution. In fact, the effect of samples from the second distribution can be filtered by subtracting $100n$ from the final result. In this way, we see that the second distribution has had no effect on the final outcome, as it is the variation, rather than the absolute value of each term, which causes a distribution to influence the final outcome.

The distributions given in this example are very similar to the distributions of energy and discomfort cost respectively; with mean equalization, the variance of the discomfort penalty distribution is significantly lower than that of the energy cost. Additionally, the repeated additions are very similar to the operations carried out by the Q-learning algorithm, which was presented earlier and will be discussed in more detail in the next section. Therefore, the method of variance equalization was chosen in order for the two cost terms to have equal effect on the optimal policy for the case of average occupancy, which was set to be equal to 12, as mentioned earlier. The occupancy level will scale the importance of comfort appropriately.

Figure 9 below shows how the variance of the discomfort penalty and energy cost (discussed in the next section) varies with the weight applied - the intersection point of the two lines is the point of interest. In this case, this weight was determined to be equal to 298.

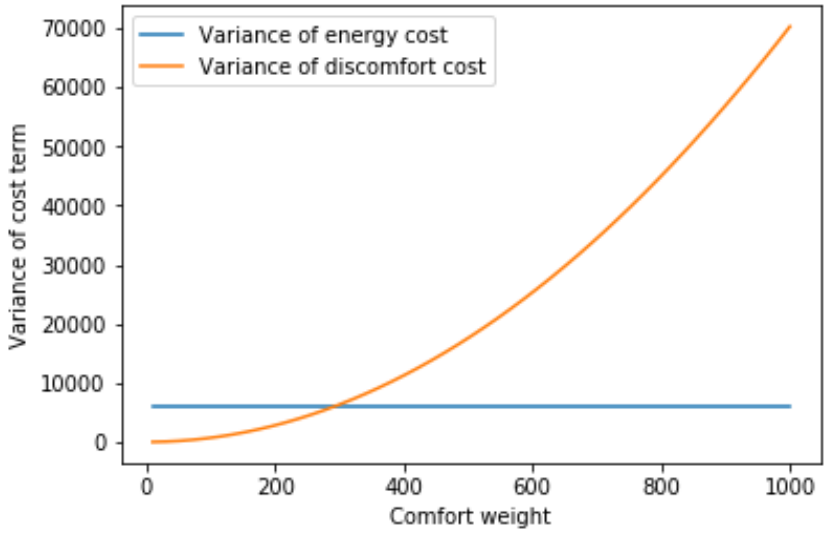


Figure 9: Variance of cost terms vs. comfort weight

3.5.2 Energy cost

The energy cost term represents the unit cost of maintaining the HVAC system at a given setpoint temperature. This function was obtained by using a multivariate linear regression model given the energy data available from the industry partner. The energy cost function is shown below:

$$E = 16.866T_{ext} + 1.4036Hu_{ext} - 23.1658T_{int} + 0.1032l_{CO_2} - 12.399T_{setpt} + 298.627$$

$T_{ext} - external\ temperature\ (^oC)$
$Hu_{ext} - external\ relative\ humidity\ (\%)$
$T_{int} - internal\ temperature\ (^oC)$
$l_{CO_2} - CO2\ level\ (ppm)$
$T_{setpt} - setpoint\ temperature\ (^oC)$

3.5.3 Cost function

Following the discussion above, the cost function is defined as follows:

$$R[s,a] = E[s,a] + n * w_c * PMV[s]$$

$R\ [s,a] - cost\ associated\ with\ taking\ action\ a\ in\ state\ s$
$E\ [s,a] - energy\ cost\ associated\ with\ taking\ action\ a\ in\ state\ s$
$n - the\ number\ of\ people\ present\ in\ the\ relevant\ region\ of\ the\ store$
$w_c - weight\ for\ discomfort\ penalty$
$PMV\ [s] - PMV\ value\ associated\ with\ ambient\ conditions\ in\ state\ s$

The values for the cost are stored in a 75 x 6 cost matrix, with the states and actions along the rows and columns respectively. Note that the energy cost is a function of the state as well as the action – from the equation given above, we see that the energy cost depends on the various components of the state (e.g. internal temperature), as well as the setpoint temperature, which corresponds to an action. The discomfort penalty is dependent only on components of the current state – although an action would alter the setpoint which would alter the comfort level of people in the store, the change in internal temperature would be delayed by at least one time step, and therefore would be accounted for in the PMV value in the subsequent time step. Since the Q-learning algorithm determined total future expected cost of taking an action in a given state, actions which cause discomfort in future time steps will be penalized.

3.6 Q-learning algorithm

As introduced earlier, the Q-learning algorithm is as follows:

$$Q_n(s,a) = (1-\alpha_n)\, Q_{n-1}(s,a) + \alpha_n[R(s,a) + \gamma * max_{a'} Q_{n-1}(s',a')]$$

We now take a closer look at the implementation of the algorithm, and the various considerations and strategies used to obtain an appropriate result.

3.6.1 Learning rate

As discussed earlier, the Q-learning algorithm requires a learning rate to be set, subject to certain constraints, in order to guarantee convergence. Initially, the learning rate was set to α = 1, to simplify the problem. However, this lead to oscillatory behaviour upon convergence, which prompted the change in learning rate to that which was eventually implemented. Figure 11 in section 4.1 shows the oscillatory behaviour of Q-values of select states during convergence when the learning rate was set to α =1.

A learning rate of $\alpha_n = \frac{1}{1+k}$ was eventually used, where k was incremented by 0.1 every time the training algorithm randomly sampled the same state as the randomly chosen initial state. For example, if the initial state was randomly chosen to be state s = 54, then k would be incremented by 0.1 every time state s = 54 was randomly sampled again. The amount by which k is incremented is at the user's discretion, but note that very small increments, which would lead to a very slowly decreasing α, can cause the Q-learning algorithm to become unstable during convergence, and even oscillate or diverge. Figure 12 in section 4.1 shows the effect of a very slowly decreasing learning rate; k was incremented by 0.005 at every iteration where the initial state was sampled again.

3.6.2 Exploration strategy

As discussed earlier, the exploration strategy by which the agent gains experience is an important part of the learning process, and can significantly affect the final outcome of the optimal policy. The initial strategy used was as follows: in each iteration of the training process, an initial state and action were chosen at random, and the Q-value for that state-action pair was updated as per the algorithm above. This method is a slight variant of what is referred to in the literature as *experience replay* (Lin, 1992), which has been used with success in recent implementations of deep Q-networks.

The idea is that the agent's various 'experiences' (s, a, r, s') are stored and randomly sampled for each iteration of training. These experiences are generally obtained from observed data, in this case temporal indoor environmental data collected by the store's sensor network. The Q-value for each state-action pair is updated, and a new experience randomly sampled for the next training iteration. A major advantage of using experience replay is to break strong naturally occurring correlations between state transitions (Mnih et al., 2013). As we saw earlier, the transition matrices showed a strong correlation with regards to state transitions; certain transitions were much more likely given certain states. Selecting consecutive observations during the learning process could lead to loops between a subset of states during training, and could cause the Q-learning algorithm to oscillate or diverge (Mnih et al., 2013). This process of learning is an example of offline learning, where the training occurs prior to actual deployment into the environment. The learning occurs using samples randomly drawn from an experience set, which is generated by historical data.

Note that information on the next state part of each 'experience' and is required at each training iteration – this corresponds to the last term in the algorithm where the maximum Q-value $Q_{n-1}(s', a')$ across all actions in the next state is added to the current Q-value. Initially, the next state s' was randomly chosen from among the subset of states with internal temperature equal to the set point temperature chosen in the previous time step, and with the same external temperature. The Q-learning algorithm was trained over thousands of iterations, with convergence occurring around 8000 iterations; however, this method of selecting the next state

did not accurately reflect the system's dynamics as it did not make effective use of the historical data – in the actual system, only a small subset of states were entered from any given state, as was seen from the transition matrices visualizations - and so the policy output by the Q-learning algorithm would not be optimal. The next section describes how this issue was resolved.

3.6.3 Modifications to algorithm and exploration strategy

To make more effective use of the available historical data, the exploration strategy was slightly modified. Specifically, the determination of the next state s' from any given state s and action a taken in state s was improved. Previously, the next state was randomly sampled from a list of all possible states. However, this would lead to an inaccurate optimal policy. Therefore, an approach which more closely resembled the actual system was required. To that end, the new determination of the next state is as follows: at each state s, an action a is taken, and the next state s' is probabilistically sampled from the subset of states that the system historically transitioned to given that particular state-action pair. The transition matrices captured exactly this information; each row corresponds to a set of probabilities of transitioning to any state s' given a current state s. The transition probabilities of each state differ based on the number of times the system made a particular transition. At every training iteration of the Q-learning algorithm, the next state is sampled according to the probabilities obtained from the transition matrix. For example, say the randomly chosen current state is $s = 23$, and the randomly chosen action is $a = 25^oC$. To determine the next state, the row corresponding to state $s = 23$ would be selected from the transition matrix for setpoint temperature 25^oC, and the next state randomly sampled with probabilities proportional to those in the selected row of the transition matrix. This method more closely resembles the workings of the actual system, and leads to an exploration of the state space which is strongly based on the system's actual historical trends.

However, there is a drawback to the method described above. If the system has never been in a particular state, then the approach above will never explore that region of the state space. If there is a zero row encountered in the transition matrix for a randomly chosen state-action pair, it means that the system never reached that state (we may make this assumption as there are no absorbing states in this formulation, leading to a zero probability that the system enters a state

and never exits). If this occurs, the row in the Q-matrix corresponding to the unvisited state will be a zero row. Upon implementation, we found that several rows (approximately 7) were zero rows. This posed a problem as the optimal policy was undefined for various states. Despite the fact that the system had never historically entered those states and so one may think they can be safely neglected, it is prudent to design an optimal policy that can handle rare events.

3.6.4 Smoothing the Q-matrix

To overcome the issue of zero rows in the Q-matrix, a somewhat novel approach was taken. The idea was to smooth out the discontinuities in the Q-matrix by assigning meaningful values to the zero rows. This was achieved as follows: after 5000 training iterations, every zero rows was assigned a value equal to the weighted average of neighbouring rows in all three directions in turn; fixed external temperature, internal temperature and occupancy level.

If smoothing could not be carried out in all three directions, it was attempted in the remaining two directions, and finally in just one direction. The hierarchy of directions, from highest to lowest, is internal temperature, occupancy level, and external temperature. For example, suppose the system has never visited state $= \{T_{int} = 26^oC, n = 13 - 19\ people, T_{ext} > 26^oC\}$, and therefore is associated with a zero row in the Q-matrix. As we can see, the internal temperature of this state is at the maximum possible value for internal temperature as per the definition of the state space. Therefore, it would not be possible to take a weighted average of the state with the same occupancy level and external temperature, but internal temperature of 1oC higher – this state does not exist in the state space. Therefore, to prevent uneven smoothing, the direction of neighbouring internal temperature is disregarded, and smoothing carried out with neighbouring occupancy level and external temperature categories. After the initial assignment of values, subsequent smoothing operations were conducted every 1000 iterations, with a different weighting – 85% of the value of the row was weighted with a total of 15% weighting across all other directions. This weighting was chosen so that the evolution of the smoothened rows was driven mostly by the value of the row itself, with small influences from other directions. The complete algorithm can be seen in Figure 10.

The Q-learning algorithm

Initialize Q_0

for $i = 0$ **to** $i = n$, **do**

$s_i \leftarrow$ Choose state

$a_i \leftarrow$ Choose action

{update Q_i}:

$$Q_i(s,a) = (1-\alpha_i)\big(Q_{i-1}(s,a)\big) + \alpha_i(R(s,a) + \gamma * max_{a'} Q_{i-1}(s',a')$$

{initial smoothing of Q_i}:

if $i = 5000$ **and** $(Q_i(s) = 0 \text{ @ } i = 5000)$ **do**

$$Q_i(s) = \tfrac{1}{6}\big(Q_i(s^{*+}) + Q_i(s^{*-})\big) + \tfrac{1}{6}\big(Q_i(s^{**+}) + Q_i(s^{**-})\big) + \\ + \tfrac{1}{6}(Q_i(s^{***+}) + Q_i(s^{***-}))$$

{subsequent smoothing of Q_i}:

if $i > 5000$ **and** $i \,\%\, 1000 = 0$ **do**

for all $Q_i(s)$ such that $(Q_i(s) = 0 \text{ @ } i = 5000)$ **do**

$$Q_i(s) = 0.85 * \big(Q_i(s)\big) + 0.05 * \left(\tfrac{1}{2}Q_i(s^{*+}) + \tfrac{1}{2}Q_i(s^{*-})\right) + \\ +0.05 * \left(\tfrac{1}{2}Q_i(s^{**+}) + \tfrac{1}{2}Q_i(s^{**-})\right) + 0.05 * (\tfrac{1}{2}Q_i(s^{***+}) + \tfrac{1}{2}Q_i(s^{***-}))$$

s^{*+} , s^{*-} : states with T_{int} 1 larger and 1 smaller than T_{int} @ $Q_i(s)$

s^{**+}, s^{**-}: states with n 1 larger and 1 smaller than n @ $Q_i(s)$

s^{***+} , s^{***-}: states with T_{ext} 1 larger and 1 smaller than n @ $Q_i(s)$

Figure 10: Modified Q-learning algorithm

4 Optimal Policy

This section presents the results of the study. Convergence of Q-values will be shown, followed by Q-matrix and optimal policy visualizations, cost comparisons and a system simulation.

4.1 Convergence of Q-values

Figure 11 shows the convergence of Q-values over 500,000 iterations with the learning rate set to $\alpha = 1$. The oscillatory behaviour seen upon convergence can be directly attributed to the poor choice of learning rate. Note that states 6 and 7 do not exhibit the same high frequency oscillations that the other two states display; these states were among the states that underwent the smoothing procedure outlined earlier. It shows that the smoothing procedure serves to provide stable and smooth values to the zero rows, whilst still tracking the general convergence trend.

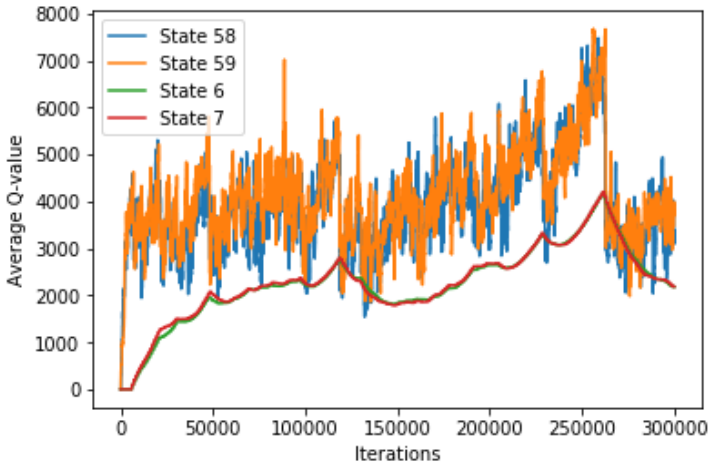


Figure 11: Convergence of Q-values of select states for constant learning rate α = 1

Figure 12 shows the convergence of the Q-values of select states over 300,000 iterations for two different learning rates of the form $\alpha_n = \frac{1}{1+k}$, where only the increment value of k differed. Note that the convergence plot on the left (k increment = 0.005) exhibits significantly less stable behaviour during convergence than the plot on the right (k increment = 0.1); as discussed earlier, a learning rate which decreases more slowly leads to less stable behaviour during convergence.

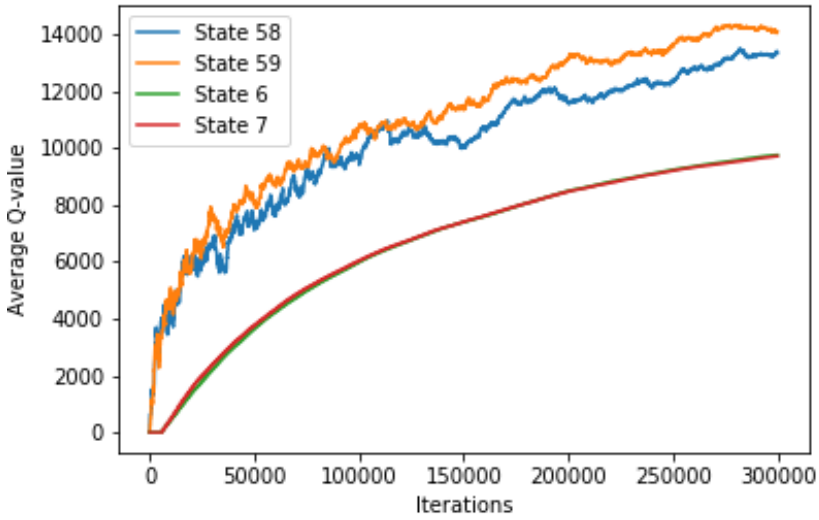


Figure 12: Convergence of Q-values of select states with k increment = 0.005

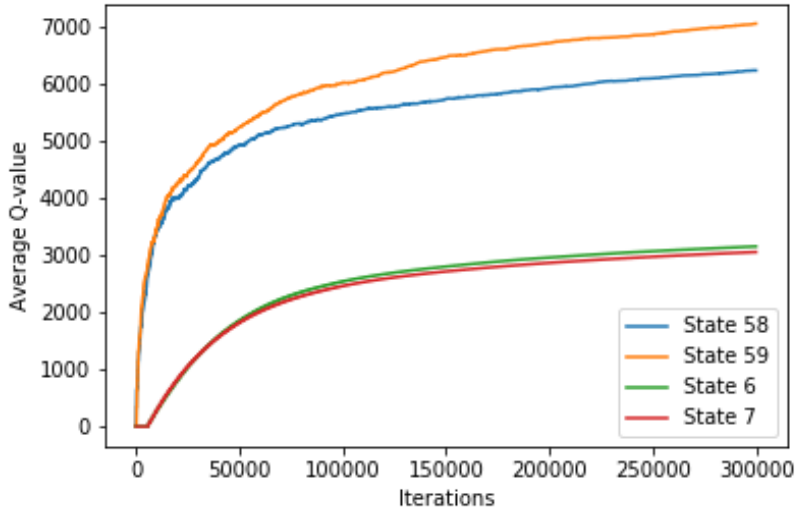


Figure 13: Convergence of Q-values of select states with k increment = 0.1

4.2 Q-matrix visualizations

Figure 13 show a visualization of a select portion of the Q-matrix. The important thing to note is that the Q-function is unimodal along the x-axis. As we can see, the Q-value consistently decreases from left to right until it reaches a minimum, from where it increases again. This is the expected behaviour; there will be a single optimal action in any given state, which will have the

lowest Q-value. Every other action in that state should have a higher Q-value, which indicates sub-optimality, and in fact should increase as the action gets further away from the optimal action. This property of the Q-matrix serves to increase our confidence in the nature of the Q-matrix.

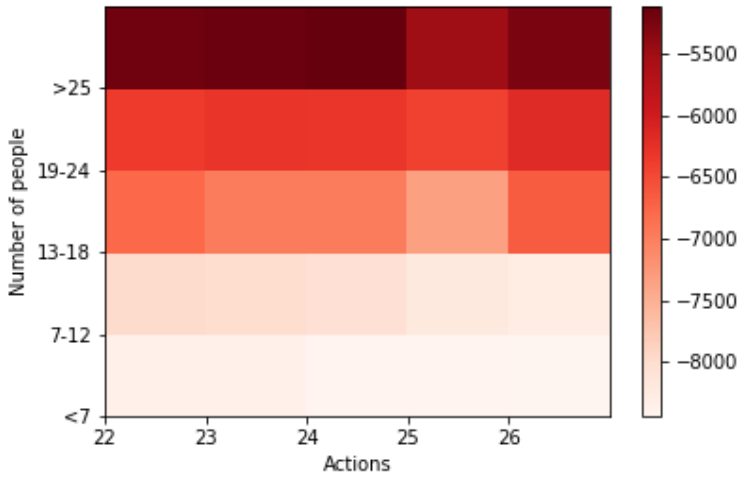


Figure 14: Visualization of select portion of the Q-matrix

4.3 Optimal policy visualizations

After training, the resulting Q-matrix captures all the information about the optimal temperature setpoint policy for the store. However, understanding the policy by looking directly at the Q-matrix is challenging as it is an abstract representation of the optimal policy. Figure 15 shows a visualization to aid with understanding the dynamics of the optimal policy. The optimal temperature setpoint is shown for all combinations of internal temperature, occupancy level and external temperature – the policy covers all possible states of the system. From the heat maps below, we see that the optimal policy mostly behaves in the expected manner. For example, the optimal setpoint temperatures for the lowest external temperature category (top left), tend to raise the temperature to 24^oC or 25^oC, and even 26^oC when there are few people present (see y axis, which shows number of people). For high external temperatures (bottom), the optimal setpoints tend to lower the temperature slightly in order to maintain customer comfort at the expense of energy savings. However, we see that several regions show optimal setpoints of 25^oC and 26^oC, which highlight the trade-off between comfort and energy savings. However, the nature of the

optimal policy is not as smooth as expected. For example, there seems to be a block of optimal setpoints of 26^oC in the central region of each map; although these setpoints may seem sub-optimal, it is difficult to judge whether or not this setpoint truly is optimal in the physical world. Bearing in mind the interactions between the energy cost and discomfort penalty in the cost function, as well as the fact that the optimal policy is determined with the aim of minimizing total expected cost over a long time horizon, it is not obvious whether these setpoints are truly optimal.

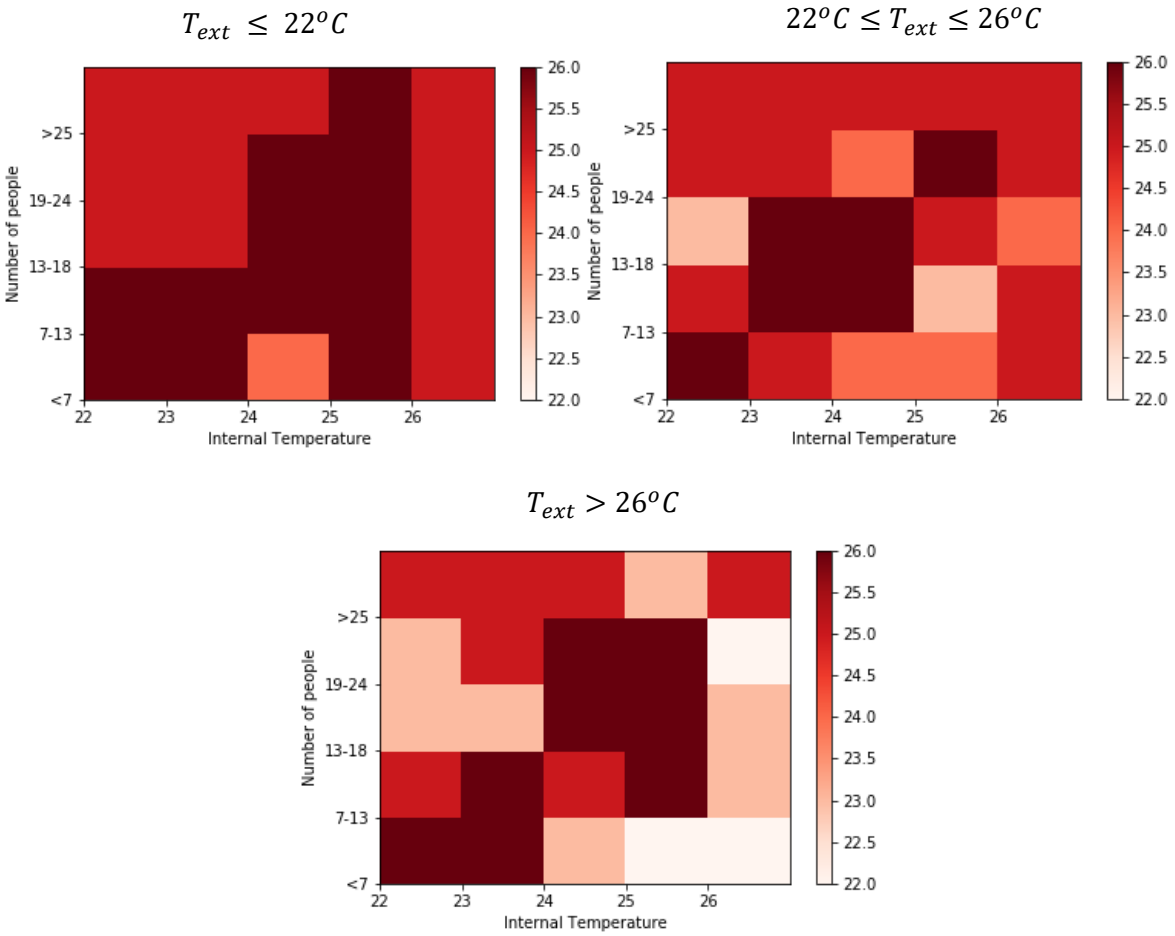


Figure 15: Heat map displaying the optimal policy

4.4 Cost comparisons

In order to determine whether the optimal policy truly is an improvement over the current policy being implemented at the store, a comparison of costs between the two policies must be made.

Total cumulative cost, as well as energy and comfort cost separately, were determined using the data set. At each time step, the cost for taking the action that was historically taken, in the state that the system was in at the time, was computed. This was carried out over the length of the entire data set, and the final costs over the time horizon were stored. The procedure was then carried out for the optimal policy; the initial state was chosen as per the historical data, the optimal action selected from the Q matrix, and the next state chosen probabilistically using the state transition matrices. This process was repeated over the same time horizon as that of the historical data. In order to determine whether the optimal policy, and therefore system cost incurred from following the policy, was stable, this entire process was repeated several times, with freshly trained Q-matrices. If the convergence of the Q-values was stable, and if the same optimal policy was obtained from multiple training rounds, then the costs should also be similar.

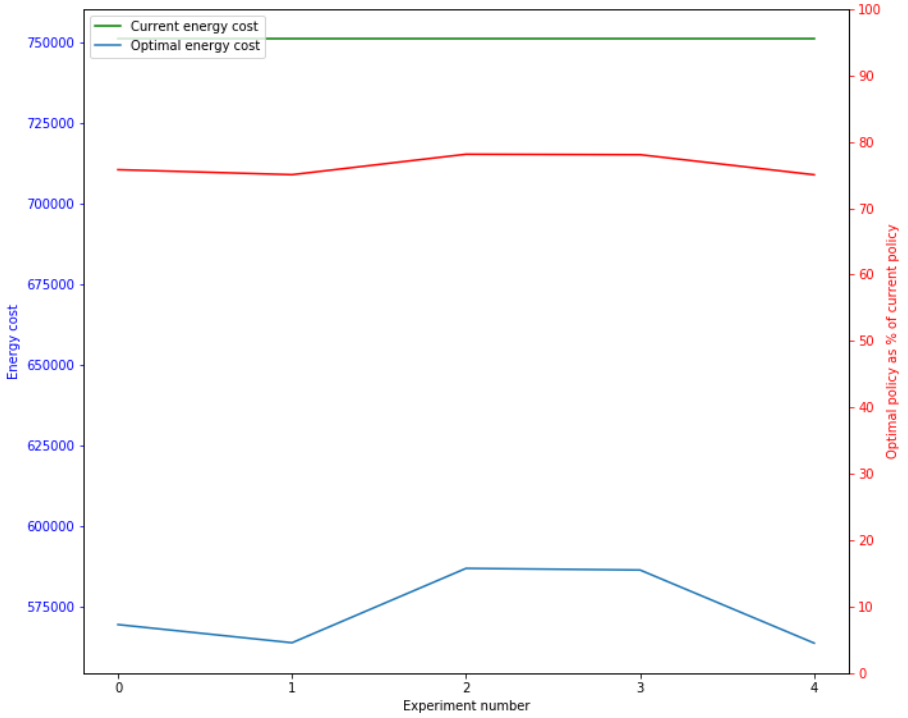


Figure 16: Energy cost comparison between the current and optimal policy

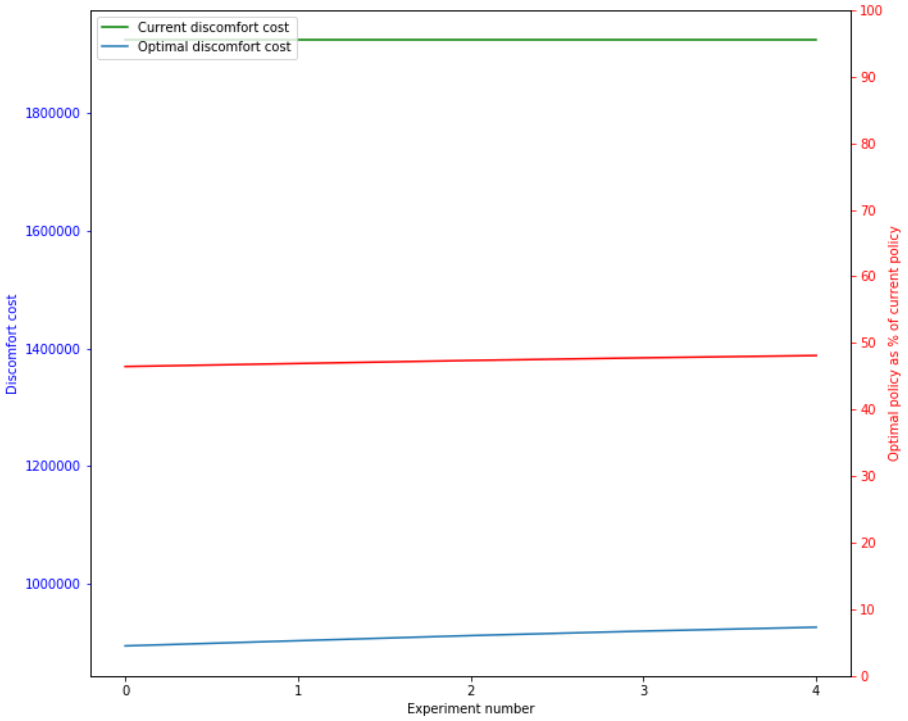


Figure 17:Discomfort cost comparison between the current and optimal policy

From Figure 18, we see that the total optimal cost is consistently lower than the total current cost, and also that the total optimal cost is similar over several experiments (~55% of the total current costs), thus confirming that the convergence and optimal policy are stable. Note that the total costs include cost of discomfort, which is not intuitive to understand when combined with energy costs. To make the cost breakdown clearer, the energy and discomfort costs have been compared separately in Figures 16 and 17. We see that the energy cost incurred from following the optimal policy is consistently lower than that incurred from following the current policy (~75% of current energy cost), and the same applies for discomfort cost (~50% of current discomfort cost).

From this we see that the optimal policy truly does provide financial savings whilst improving customer comfort.

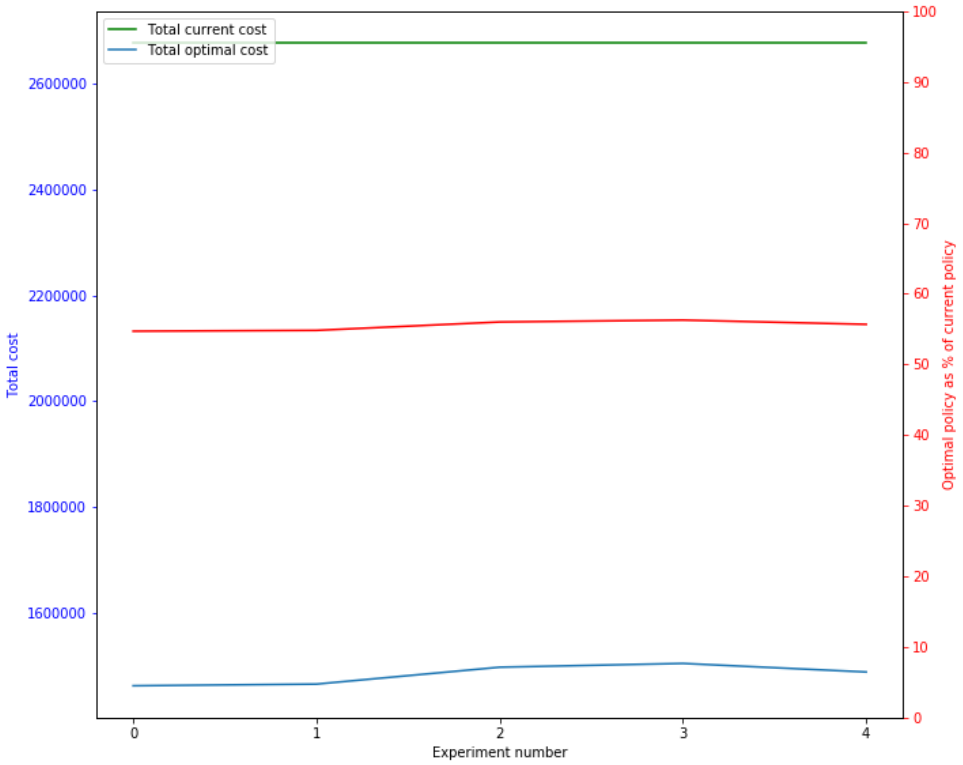


Figure 18: Cost comparison of total combined energy and discomfort costs between current and optimal policy

4.5 System simulation

Figure 19 shows a system simulation over 1000 time steps as an additional visualization tool for understanding the general dynamics of the optimal policy. The simulation was conducted in the same way as the optimal policy cost determination, where an initial state was chosen, and the optimal policy followed as the system evolved over 1000 time steps. This simulation serves to illustrate the most common action selections by the optimal policy over 1000 decision steps. It can be observed that the most common optimal actions (i.e. red dots) were temperature set points of 23ºC, 24ºC and 25ºC. This supports the intuition that the optimal policy will take actions closer to the midpoints of the action space, rather than the extremes, for which it may incur larger energy or discomfort costs, both directly but also as subsequent actions would incur heavier costs for bringing the internal temperature to a desirable setpoint.

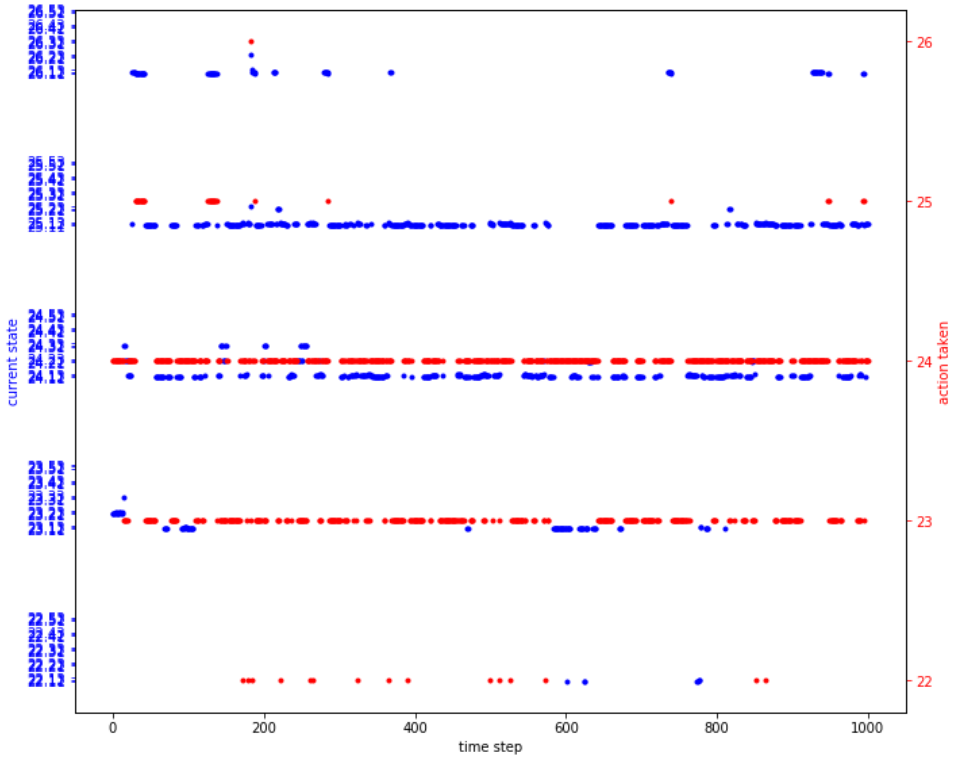


Figure 19: System simulation over 1000 time steps

5 Further Work

5.1 Extension to multi-agent system

The study is to be extended to the multi-agent system which includes all 3 HVAC systems. All the work to date has been carried out on a single HVAC system (HVAC 2) and will yield the optimal policy for HVAC 2, but the ultimate aim is to determine a global optimal policy across all 3 systems. The current implementation is extendable to each of the three individual HVAC systems.

An extension of the Q-learning algorithm to multiple agents is based on the idea of pairwise equilibrium. The algorithm starts with a pair of adjacent agents, each responsible for an HVAC, and tries to find an equilibrium between the two agents. To this end, the algorithm finds an

optimal policy for one agent as an optimal response to the optimal policy of the other agent. Once an equilibrium is found, we switch to a new pair consisting of an agent from the current pair and a new agent. When an equilibrium is found between the chosen pair, we repeat the procedure for the other pair in which one agent corresponds to HVAC 2 (the centre HVAC region), until convergence is obtained. The reason for the choice to always include HVAC 2 in each pair is to reduce the complexity of the problem; we assume that interactions between HVAC 1 and HVAC 3 will be minimal as HVAC 2 serves as a buffer between the two systems. Therefore, the optimal policy of HVAC 2 will affected by the optimal policies of both HVAC 1 and HVAC 3.

In the two-agent pairwise Q-learning, the state definition should be augmented so that each agent should be able to take actions conditional on the state of the two-HAVC system as well as the action to be chosen by the other agent in the given state. Therefore, the state definition should be as follows:

$$S = \{(t_{internal,1}, t_{internal,2}, n1, n2, t_{external}, a) \mid t_{internal,i} \in T_{internal}, t_{external} \in T_{external}, n_i \in N, a \in A\}$$

where $t_{internal,k}$ and n_k are the internal temperature and the number of shoppers in the section controlled by agent k $(k = 1,2)$ respectively, and a is the action taken by the other agent in the pair. Note that external temperature is common for both agents and denoted as $t_{external}$.

5.2 Population estimation

Given the inaccurate output from the population estimation model used, it may be valuable to investigate other methods of population estimation. Methods based on CO2 measurements may be sufficient, if slightly different methodology is used to convert between the measureents and estimated occupancy level. For example, real world observations about the average number of people actually seen in the store at certain times, say 10am and 7pm, could be used to match those observed values with the CO2 measurements taken by the sensors. Matching the minimum CO2 concentration, say in the early hours of the morning (e.g. 4am) with 0 occupancy, and scaling the intermediate values, either in a linear fashion or using a polynomial fit, may increase the accuracy of the occupancy estimation portion of the study.

5.3 Heat transfer and dynamic HVAC models

Improvements in the modelling of the underlying physical processes taking place can serve to improve the results, especially when extending the study to the multi-agent system. Developing a model of the heat transfer characteristics of the store between adjacent HVAC systems can better inform each agent's optimal policy during pairwise training. Additionally, further development of the model for the HVAC systems would be useful in obtaining better results. An important consideration that can be included in the determination of costs can include 'inertia' of the HVAC system i.e. additional costs incurred by repeated altering of setpoint temperature, especially when decreasing the setpoint temperature. It is entirely possible, for example, that bringing the internal temperature from 26 to 24 repeatedly would incur higher costs than maintaining the temperature at a lower temperature, say 23. This hypothesis is based on the fact that motors have a non-linear power draw characteristic; for an induction motor, the current drawn from the supply varies non-linearly with the applied load (Allen Bradley). Motors are also known to have load regions where performance is optimal, and so moving in and out of such a region could lead to inefficient power use. Therefore, a reward function which considers such factors may lead to a policy in which repeated setpoint changes are discouraged. This could have an interesting effect on the optimal policy.

5.4 Exploration strategies

We have seen that the exploration strategy used in this study led to certain regions of the state space remaining unexplored. This was remedied by carrying out smoothing operations, but a more systematic solution to this issue may yield improved results. An example of a solution to the issue of unexplored regions is to assign very low, specifically non-zero, values to entries in the transition matrices, thus rendering them less sparse than they currently are. The value assignments can also monotonically decrease from the line of maximum correlation seen in the visualizations of the transition matrices. These small non-zero values will introduce a low probability of visiting certain states that were never visited historically by the actual system. During training, probabilistically sampling the next state will now include a probability that these

previously unvisited states may be simulated. This approach may well eliminate the zero rows seen in the Q-matrix prior to smoothing.

5.5 Alternative approaches

The field of reinforcement learning extends well beyond the Q-learning algorithm used in this study. Early advances in the field introduce, amongst others, the idea of learning by the method of temporal differences (Sutton, 1988); arguably, Q-learning is a specific variation of such learning methods, but is certainly not the only one. As such, other methods which solve reinforcement learning tasks may be worth investigating. The general method of learning by temporal differences introduced by Sutton (1988) involves learning the trends of the historical data in order to predict future behaviour. At the core of the learning algorithm lies a value iteration, with the key difference that optimization is not carried out at each iteration, but rather after the initial learning process. Using this method, the algorithm can learn the historical data by repeatedly sampling, with replacement, individual days from the data set (this process is known as bootstrapping, and is similar in concept to experience replay encountered earlier). After learning is complete, the output will contain information on the total expected reward of taking certain actions in certain states, and this information can be used as the basis for forming an optimal policy.

6 Conclusion

In this study, we developed a variant of the Q-learning algorithm and implemented it in order to determine an optimal temperature setpoint policy for the retail store. An optimal policy was determined, and the results for the single agent system showed that the optimal policy reduced the store's energy cost by approximately 25%, and discomfort cost by approximately 50%. The study is to be extended to the multi-agent system, and notes on further work are provided.

References

Allen Bradley. (n.d.). Drive and motor basics. Allen Bradley support.

American Society of Heating, Refrigeration and Air Conditioning Engineers. (2013). ASHRAE 55 - Thermal environmental conditions for human occupancy.

ASHRAE. (2003). Ventilation for acceptable indoor air quality - Addendum n to ASHRAE standard 62-2001.

Beynier, F.-I. A. (2010). Decentralized MDP's. In *Markov Decision Processes in Artifical Intelligence.* Wiley.

Emmerich, S., & Persily, A. (2001). *State of the art review of CO2 demand controlled ventilation technology and application.* National Institute of Standards and Technology.

Fanger, P. (1970). *Thermal Comfort: Analysis and applications in environmental engineering.* Danish Technical Press.

Garcia, F., & Rachelson, E. (2010). Markov Decision Processes (Introduction). In *Markov Decision Processes in AI.* Wiley.

Intel. (2016). *Smart buildings with internet of things technology*. Retrieved from https://www-ssl.intel.com/content/www/us/en/smart-buildings/overview.html.

Ito, S., & Nishi, H. (2012). Estimation of the number of people under controlled ventilation using a CO2 concentration sensor. *38th Annual Conference on IEEE Industrial Electronics Society.*

Leephakpreeda, T. (2001). Occupancy based control of indoor air ventilation; a theoretical and experimental study. *ScienceAsia*.

Lin, L.-J. (1992). Reinforcement learning for robots using neural networks . PhD Dissertation, Carnegie Mellon University.

Mnih et al. (2013, December). Playing Atari with Deep Reinforcement Learning. DeepMind Technologies.

Savitsky, A., & Golay, M. (1964). Smoothing and differentiation of data by simplified leasat squares procedures. *Analytical Chemistry*, Volume 36, Issue 8, pp. 1627-1639.

Shift Energy. (2016). *EOS Architecture; Intelligent Energy Management*. Retrieved from http://shiftenergy.com/eos-architecture/

Sutton, R. S. (1988). Learning to predict by the method of temporal differences. *Machine Learning Volume 3, Issue 1*, 9-44.

Wang et al. (1999). Experimental validation of CO2 based occupancy detection for demand controlled ventilation. *Kargerr, 8*.

Watkins, C. (1989). Learning from delayed rewards. *PhD Thesis*, 279-292. Imperial University, London, England.

Watkins, C., & Dayan, P. (1992, May). Technical note: Q-Learning. *Machine Learning 8 (3-4)*, pp. 279-292.

Yang, N. -G. (2013). A systematic approach to occupancy modeling in ambient sensor-rich buildings . *SAGE journals, 90*.