Model-Free HVAC Optimizer based on Reinforcement Learning

Charalampos Marantos
Department of ECE
National Technical University of Athens
Athens, Greece

Christos Lamprakos
Department of ECE
National Technical University of Athens
Athens, Greece

Kostas Siozios
Department of Physics
Aristotle University of Thessaloniki
Thessaloniki, Greece

Dimitrios Soudris
Department of ECE
National Technical University of Athens
Athens, Greece

2023 IEEE 32nd International Symposium on Industrial Electronics (ISIE) | 979-8-3503-9971-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISIE51358.2023.10227979

***Abstract*—Recently, there is a continues demand for embedded systems that automate buildings' operation, such as the control of Heating Ventilation and Air-Conditioning system (HVAC) operation. These systems exhibit increased complexity and their operation relies less on human decision-making and more on computational intelligence. The efficiency of these systems is usually bounded by the orchestrators' flexibility to optimize simultaneously multiple, and usually contrary, objectives. This paper introduces a novel framework for designing model-free orchestrators targeting to optimize the operation of HVAC systems, is introduced. The proposed orchestrator relies on Reinforcement Learning in order to support self-adaptive customization. Experimental results highlight the superiority of introduced orchestrator, as it achieves comparable performance to state-of-the-art relevant controllers without any prior detailed modeling.**

***Index Terms*—Model-Free Optimization, Multi-Objective Optimization, Smart Thermostat**

I. Introduction

Buildings are increasingly energy-demanding and it is expected to consume even more in the future. The amount of energy consumed in European Union's buildings reaches around 40—45% of the total energy consumption, where two-thirds of this energy is used in dwellings [1]. More specifically, Heating, Ventilation and Air-Conditioning (HVAC) is the largest contributor to the building's energy cost, as they are rarely configured with a static (non-optimal) way. On top of that, residents adjust manually the thermostats several times a day, while programmable thermostats are too difficult for the majority of people to be used effectively. To make matters worse, it is typical households with programmable thermostats to have higher energy consumption on average than those with manual controls because users program them incorrectly or disable them altogether.

Moreover, as the physical world is bound by unpredictability, it is not prudent to expect the HVAC control system to be operating in a fully-controlled environment; thus, solutions that rely on *robust* and *self-adaptable* to unexpected conditions controllers are of upmost importance. Formally, self-adaptivity refers to systems that adjust their behavior autonomously in response to external events unpredictable, or unforeseen, at design time [2]. The orchestrators' selections are usually defined according to the HVAC system's cost function, which considers multiple objectives that have to be compromised simultaneously. Hence, novel solutions able to perform optimal configuration of HVAC systems are upmost necessary.

Recently, a new generation of systems with integrated computational and physical capabilities, also known as CyberPhysical Systems (CPS), have been introduced. The new design paradigm interacts with, and expands the capabilities of, the physical world through monitoring, computation (i.e., distributed coordination) and communication mechanisms. By pooling the system's resources and capabilities together results to a new, more complex system which offers additional functionality and performance than simply the sum of the constituent sub-systems. The CPS approach offers a better resolution of the physical world and therefore a better capability of detecting the occurrence of an event; hence, it is expected to play a key role in the development of next-generation autonomous systems, especially for the smart building environment.

Although promising, the efficiency of these HVAC control mechanisms relies mainly on the employed algorithms that perform system's orchestration under real-time constraints. Existing approaches towards this direction rely on computational intensive decision-making algorithms that are executed onto powerful processing core(s). In accordance to the previously mentioned challenges, throughout this paper we introduce a general-purpose framework for HVAC orchestration, where two conflicting objectives, namely the overall energy consumption and the occupants' thermal comfort have to be compromised at run-time. In contrast to relevant solutions, the proposed orchestrator focuses on partial, or fully unknown, cost functions.

This research has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH-CREATE-INNOVATE (project code: T2EDK-01681).

II. Related Work

The HVAC system's orchestration is a well-established challenge that has been extensively analyzed. Despite the significant progress made in optimal nonlinear control theory [3] [4], existing methods are not, in general, applicable to large-scale systems because of the computational difficulties associated with the solution of the Hamilton-Jacobi partial differential equations. According to literature there are two mainstream ways for deciding upon HVAC system's orchestration. In detail, the first category corresponds to systems that provide online decision-making [5], while the latter approach relies on Model Predictive Control (MPC) techniques [6] [7]. Although MPC for nonlinear systems has been successfully applied in various domains [6] [8] [9], it likewise encounters dimensionality issues: in most cases, predictive control computations for nonlinear systems amount to numerically solving a non-convex high-dimensional mathematical problem [10], whose solution may require formidable computational power for supporting online decision-making. Regarding the micro-grid case study, the online algorithms exhibit limited efficiency compared to MPC solvers, but they are reactive to real-time constraints (e.g., indoor/outdoor temperature and humidity, solar radiation, occupants behaviour, state of neighboring thermal zones, etc) [7] [5]. On the other hand, the efficiency of MPC solvers relies on a detailed model of the target system to simulate the impact of alternative control strategies [6]. The fact that the detail system's modeling is too complex coupled with the need of the MPC algorithms to be delivered prior to the system's deployment create problems for building low-cost model-free solutions.

Although the aforementioned techniques are in-line with the concept of a model-free HVAC orchestrator discussed throughout this manuscript, they exhibit limited flexibility. Specifically, both supervised and unsupervised machine learning algorithms impose excessive training complexity, which cannot be tackled with a low-cost embedded platform (i.e. a smart thermostat) under *run*-time constrains. Similarly, fuzzy rules cannot "*learn*" effectively (e.g. in terms of accuracy and execution run-time) the CPS's behavior for unexpected operating conditions. Consequently, a pre-training phase per case study (e.g. type of building, different HVAC systems, variations at weather conditions, etc) is necessary. To address this challenge, another class of decision-making algorithms, also known as Reinforcement Learning (RL), have been proposed. In detail, based on RL decisions, the orchestrator take actions to maximize some notion of cumulative reward. During the last years, there us a continues demand for solvers that aim to address the HVAC system's orchestration with RL algorithms [11] [12] [13] [14]. However, none of them introduces a model-free approach, similar to the one discussed throughout this manuscript.

III. Proposed Model-Free HVAC Orchestrator

This section introduces the template of our case study, which corresponds to a micro-grid environment, which includes multiple energy sources (e.g., solar, wind, bio-gas) and nodes that are in need of energy, such as the HVAC systems. In order to support the HVAC control task, a number of sensors acquire weather data (i.e. temperature, humidity and solar radiation), building conditions (indoor temperature and humidity), as well as the residents activity per thermal zone. This data is transferred to the main controller in order to compute optimal actions that co-optimize thermal comfort and energy cost metrics.

A. Problem Formulation

Throughout this Section we discuss the analytical form of the HVAC optimization problem. For this purpose, we model a multi-objective optimization problem (MOO), formally defined with Equation 1, where E and PPD give the two objectives (energy consumption and occupants' thermal comfort) under minimization. PPD corresponds to the Predicted Percentage of Dissatisfied occupants, while $PPD \leq PPD_{limit}$ gives the thermal comfort constraints that have to be satisfied. At this notation, α_i is a the input variables that refers to the temperature set-point of the target HVAC system in time-step i. Finally, we consider that objective functions are also related to an external vector of environmental variables s_i. The proposed framework focuses to a subset of the general MOO problem, where the cost function is expressed as a weighted sum of single objectives [15]. However, the increased complexity of contemporary buildings makes it prohibiting, or even impossible, to consider at Equation 1 an accurate analytical description while providing a Plug&Play solution; thus, the definition of the problem that this manuscript aims to solve is given by the Equation 1 coupled with the following properties:

1) the detailed form of the objective functions (E and PPD) is unknown;
2) the objective functions are not only related to the temperature set-point, but also to the buildings environment;
3) by considering discrete time/events (time-steps), the controllers actions are evaluated once per time-step;
4) the range of the Energy consumption (E) is unknown, as no prior information is given.

$$\begin{aligned} Minimize: tr \times E(\alpha_i, s_i) + (tr-1) \times PPD(\alpha_i, s_i) \\ subject\ to: PPD(\alpha_i, s_i) \leq PPD_{limit} \end{aligned} \quad (1)$$

In detail, according to the first property, the HVAC optimization problem *cannot be analytically defined at design time*; thus, a self-adaptive method is necessary. Furthermore, the cost function is affected by building's environment *enlarging the problem's complexity*. The third property corresponds to the "*penalty*" of exploring the solution space. In the context of this manuscript, the end-user is responsible for deciding upon the weights (trade-off) selection. The proposed framework provides proper automatic normalization of the employed objectives in order to respect user's preferences, as they are expressed with the selected weights. Note that this requirement is inline with weighted-sum optimization techniques found in relevant literature [15], where authors identify the necessity

for normalization of objective functions in order to achieve *adherence to the preferences of the designer*.

B. Reinforcement Learning Algorithm

The proposed Reinforcement Learning algorithm consists of a set of states, a set of actions, and a reward function. In RL terminology, we call *Agent* the mechanism that learns and acts. *State* is a vector that describes the scenario/situation that the Agent encounters in the Environment. The Agent performs an *action* that has an effect on the environment and therefore changes the *state*. Every *action* gives a *reward* from the environment to the agent; in this context, the agent's objective is to maximize its total reward during an episode (i.e. until reaching a *terminal state* and start a new *episode*). Finally, the feedback from a terminal state is zero, or even a negative reward.

During algorithm's execution, an action a_i (we assume that the action space is finite) is selected iteratively once per instance leading from state s_i to a new state s_{i+1}. At this notation, the tuple (s_i, a_i, s_{i+1}) is called a *transition* and a real reward value r_i is assigned to each of them. The agent's objective is to find a series of transitions that maximize the total return reward value (also called the return r') in Equation 2. Finally, the $\gamma \in [0, 1)$ is a discounting factor that controls the importance of future rewards and ensures convergence of the sum in Equation 2.

$$Maximize\ r' = \sum_{i=0}^{n} \gamma r_i \tag{2}$$

Given a state s_1 and an action a_1, the action-value of the pair (s_1, a_1) is defined by Equation 3, where r' is a random return associated with first taking action a_1 in state s_1, thereafter. Consequently, the estimation of Q parameter is of high-importance, as it quantifies the efficiency of CPS orchestrator's selections.

$$Q(s_1, a_1) = \mathbb{E}[r'|(s_1, a_1)] \tag{3}$$

C. Proposed framework

This section describes the proposed framework for addressing the model-free HVAC optimization challenge. In detail, this framework performs the dynamic control of the HVAC system in predefined intervals, called *time-steps*, by sampling the state of the building and computing the next *set-point*, i.e. a vector designating the values of all the configuration parameters of the HVAC system.

In order to address the inherent unpredictability of the unknown underline system, the proposed framework relies on a Reinforcement Learning (RL) algorithm [16] to: (i) navigate between available system's states, and (ii) model and predict cost function according to the already acquired (from the CPS sensors) data. The studied RL algorithm models the target system by means of a set of *states*, a set of *actions* and a reward function. Since the reward maximization is equal

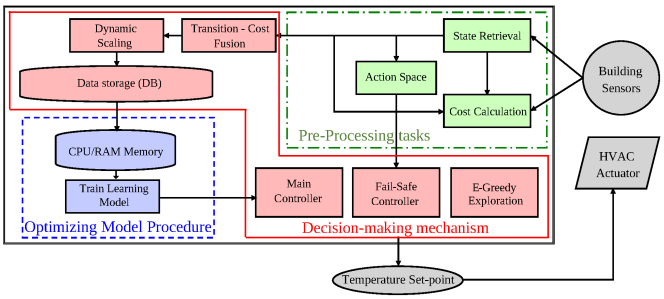


Fig. 1: Proposed framework for design and customization of the targeted model-free HVAC orchestrator.

to the minimization of cost function (i.e. by considering the reward presented in Equation 4), for the rest of our analysis we will refer to c as the *cost* function. At this notation, a real value c_i is assigned per transition (per time-step). Estimating the action-value function Q (based on Equation 3) is of high-importance for evaluating the overall orchestrator's performance. Moreover, as the employed costs c_i are unknown, a function approximation (by means of data-driven supervised learning) is applied.

$$r_i = MAX\_COST - c_i \tag{4}$$

1) Pre-processing tasks: The functionality of these tasks is to guarantee that the preceding decision-making logic can dynamically respond to unexpected events. For this purpose, the self-adaptive mechanism is invoked.

State retrieval: It retrieves the current buildings' state once per time-step. In order to guarantee effective Q-function approximation (based on Equation 3), each state has to fulfill the Markovian property, i.e. to contain all the necessary information for estimating the Q function. Therefore, the state is formed by the subset of acquired data that influence system's functionality. At this notation, a system's state (Equation 5) is defined as a tuple of (*Outdoor temperature (T_i^{out}), Solar radiation (R_i), Indoor temperature (T_i^{in}), Indoor humidity (H_i)*), since they effectively capture the building's dynamics for both energy consumption and thermal comfort metrics.

$$s_i = [T_i^{out}, R_i, T_i^{in}, H_i] \tag{5}$$

Action Space: For each time-step, an action refers to the assignment of a temperature set-point per thermal zone. Then, the action space accrue by dividing the range of HVAC's set-point (temperature) into discrete segments. A step-by-step search approach is applied for this purpose. Regarding the studied problem formulation, the required action space a_i includes five candidate options with respect to the current temperature, i.e. maintaining current temperature, increase/decrease it by a step of $\pm 0.5°$C or $\pm 1.0°$C degrees (Equation 6).

$$\alpha_i \in \{T_i^{in} - 1, T_i^{in} - 0.5, T_i^{in}, T_i^{in} + 0.5, T_i^{in} + 1\} \quad (6)$$

Cost calculation: The cost function is formed by a weighted sum of the HVAC objectives (energy cost and thermal comfort). Adopting the approach from [17], with respect to the constraints that must be satisfied (Equation 1), an action is deemed *terminal* (i.e., corresponds to a terminal cost of value max_cost) if it leads to prohibitive results regarding the constraints (thermal comfort), or the constraints were already unsatisfied and the action further increases the dissatisfaction value. Our framework evaluates the efficiency of applied, or candidate actions, through the cost function given at Equation 7. The weighting factor ($tr \in [0, 1]$) gives the relative importance between the two orthogonal metrics, namely the *energy cost* and the occupants' *thermal comfort level*.

$$c(s_i, \alpha_i) = \begin{cases} tr \times E_{norm}(s_i, \alpha_i) + (1 - tr) \times PPD_{norm}(s_i, \alpha_i), & PPD(s_i, \alpha_i) \leq 15 \\ max\_cost, & \text{else} \end{cases} \quad (7)$$

Regarding the energy cost (E), if the expected energy loads of the buildings at time-step i exceed the energy availability from the photovoltaic panels (PVs), the excessive demand is met by purchasing additional energy from the main-grid at price. Otherwise, the energy requirements are met with micro-grid's renewable sources.

Given that the value of thermal comfort (PPD) cannot exceed 15[1], the cost metric at Equation 7 is not terminal whenever $|\text{PPD}| \leq 15$. On the other hand, an action is deemed terminal (i.e., corresponds to a terminal cost) if:

- the thermal comfort metric is out of acceptable range ($PPD > 15$);
- the thermal comfort metric was already out of the acceptable range and the action further increases its value.

2) Decision-making mechanism:: This subsection describes the functionality of the introduced RL algorithm by taking advantage of its acquired knowledge to increase the efficiency of the predicted actions. In principle, this is a leap towards self-adaptive orchestrators, as any unforeseen and manifested condition will be considered at the supervised machine learning model.

Transition-cost fusion: The controller's efficiency is based on the history of all encountered states, taken actions, calculated costs and the preceding states as a result of these actions. A batch of data in the form of concatenated tuples (s_i, a_i, c_i), one per transition (s_i, a_i, s_{i+1}), is created and stored to the database to support the learning procedure.

Dynamic scaling: As stated at Section III-A, an accurate scaling (normalization) of the objective functions must be accomplished especially for the energy consumption, where the range is unknown without prior information about the buildings' dynamics. Our framework lies to *unsupervised dynamic scaling* [18], where running average and standard deviation are calculated for all the objectives. As new data acquired from building's sensors, the scaling parameters are re-calculated, ensuring that the scaling is up-to-date. Equation 8 formulates the normalization of function F, where μ^i and δ^i denote the current (i-th time-step) mean value and the standard deviation, respectively. Similarly, Equations 9 and 10 give the iterative update of μ^i and δ^i values.

$$F_{norm} = \frac{F - \mu^i}{\delta^i} \quad (8)$$

$$\mu^i = \mu^{i-1} + \frac{F - \mu^{i-1}}{i} \quad (9)$$

$$\delta^i = \sqrt{\frac{\sigma^i}{i-1}}, \quad \sigma^i = \sigma^{i-1} + (F - \mu^{i-1})(F - \mu^i) \quad (10)$$

Data storage: Our framework considers that both transitions and costs are stored in databases to enable model's refinement task.

Main controller: Given the current state and the available actions, the orchestrator designates the configuration for the next set-point, which minimizes the expected return (i.e. the cumulative future costs in the time frame defined by γ in Equation 2).

An Artificial Neural Network (ANN) is used for supporting the task of machine learning. The training of this ANN is performed with the database in the form of (s_i, α_i) tuples. The targets for this training are computed based on Equation 11, where Q_i denotes the output of the *current* ANN. Finally, the term $c_i(s_i, \alpha_i)$ is a linearly scaled value of the actual cost as a function of cost definition and dynamic scaling adjustment.

$$target = c_i(s_i, \alpha_i) + \gamma \cdot min_{\alpha_i} Q_i(s_{i+1}, \alpha_i) \quad (11)$$

Fail-Safe controller: When terminal cost is exceeded, the following two procedures take place: Initially, the *fail-safe controller* is temporarily invoked to select the next set-point and then the *learning model* is re-trained. The fail-safe controller selects an action, which exhibits increased probability to improve the overall solution in term of thermal comfort constraints satisfaction. It merely rectifies the previous "sub-optimum" action and based on previous observations it computes a set-point that ensures thermal comfort constraint satisfaction. More precisely, the actions of the fail-safe controller increase or decrease the indoor temperature by 1°C, according to the residents estimated thermal comfort.

ϵ-greedy exploration: In order for the proposed framework to solve optimally non-convex problems, an exploration mechanism for avoiding being trapped in a local minimum is necessary. In this work, we propose an ϵ-greedy exploration mechanism for avoiding this pitfall. The main controller operates with possibility $1 - \epsilon$, where ϵ is self-regulated (Self-Regulating Action Exploration (SRE)) [19]. The value of ϵ

[1]This threshold is defined as the acceptable limit according to EN15251 European standard.

is calculated based on the orchestrator's success rate and its update is performed by Equation 12, where λ and k are the success rate and learning rate, respectively.

$$\epsilon' = f(1-\lambda)\Big[k(1-\lambda)+(1-k)\epsilon\Big], where\ f(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (12)$$

The λ parameter is based on the orchestrator's consecutive positive outcomes, which are determined by the validity of the learning model's prediction. Inspired by [20] we propose a definition of this validity that is associated by the Temporal Difference (TD) error formulated by Equation 13. Specifically, controller's decision is deemed positive if $|TD| \leq TD\_limit$, where TD_limit has to be small enough to represent an accurate approximation and elastic enough to encourage early stages of learning.

$$TD = c_i(s_i, \alpha_i) + \gamma \cdot min_{\alpha_i} Q(s_{i+1}, \alpha_i) - Q(s_i, \alpha_i) \quad (13)$$

Table I summarizes the parameters employed within the proposed model-free orchestrator. At this table we also provide the values of these parameters regarding the studied case study (i.e. efficient configuration of HVAC system at agnostic buildings towards improving thermal comfort and energy savings metrics).

3) Optimizing Model Procedure:: This task performs the iterative model optimization. Specifically, it deals with the data transfer from system's database to RAM or GPU memory, the ANN's targets calculation according to Equation 11, as well as the ANN retraining.

Train Learning Model: Two different approaches for ANN retraining are considered. The first of them assumes that one training epoch is performed once per time-step, whereas the latter approach concerns a full ANN retrain in case a terminal-cost is reported. Since an ANN retraining includes many epochs, it exhibits increased computational complexity and execution run-time overhead. During the initial time-steps, ANN retrain is performed more frequently due to the limited orchestrator's "knowledge"; however, the reduced amount of training data imposes negligible overhead for this task. On the other hand, the retrains for an already trained RL algorithm are limited; hence, the associated overhead (due to the increased database size) is also limited.

IV. Experimental Results

The problem we tackle throughout this manuscript deals with the optimum configuration of HVAC systems in order to maximize occupants' thermal comfort with the minimum possible energy cost. The target case study considers five buildings with multiple thermal zones (summarized at Table II), while the efficiency of the proposed orchestrator is evaluated with the usage of well-established EnergyPlus suite [21]. The buildings' modeling was performed in detailed manner[2] [22]. Regarding the employed weather and energy pricing data, they correspond to publicly available information for 2010 [23] [24]. Without loss of generality we consider that both energy consumption and thermal comfort metrics are of equal importance; thus, the selected weights at Equations 1 or 7 are equal to 0.5.

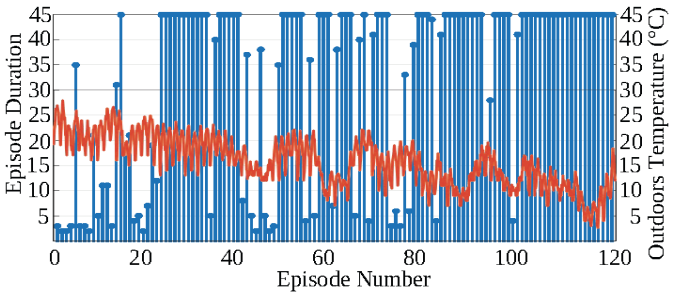


Fig. 2: Evaluation of the orchestrator's performance over time: Episodes duration regarding the January–March experiment.

The targets of the proposed data-driven machine learning method aim to derive temperature set-points that not only lead to lower costs for the current time-step, but also optimize future orchestrator's decisions. For this purpose, careful selection of the γ parameter (Equation 11) is necessary. Based on our exploration, we conclude that the performance of proposed orchestrator is retrieved for γ equals to 0.98. The estimation of objective functions is performed via a Multiple Layers Perceptron (MLP) ANN. The activation function for all the nodes except those of the output layer is the hyperbolic tangent sigmoid, whereas the output layer incorporates the logistic sigmoid to produce values in the range $[0, 1]$. Similarly, the λ parameter for the *ϵ-greedy estimation* component refers to the number of successful actions of the controller. For the scopes of this manuscript, according to the required characteristics of the TD_limit presented in Section III-C2, we consider an action is valid if the MLP's $|TD|$ is less or equal to 0.15 (Equation 13). With regard to data manipulation for local database storage, we selectively save the data that refer to the last N days in order to reduce the incremental batch of data. Such a technique has proven to be very efficient in use-cases like the one examined on this paper [25].

A. Evaluate Orchestrator's Efficiency

Initially, we quantify the efficiency of the proposed framework in terms of *episodes*. An episode is a sequence of control iterations, that ends if the current state fulfills a termination condition (e.g. the system reached its goal state, or a failure occurred) [26]. In the context of this case study, an episode ends either when the orchestrator results to thermal comfort values out of the acceptable limits, or the HVAC system is turn off at the end of a day.

The results of this analysis are depicted in Figure 2. For demonstration purposes, the horizontal axis plots the id of

[2]The modeling of the buildings was part of the PEBBLE FP7 project funded by the European Commission under the grand agreement 248537.

TABLE I: Summary of the studied problem and the proposed orchestrator's definition.

State	$s_i = [T_i^{out}, R_i, T_i^{in}, H_i]$	
Action Space	$\alpha_i \in \{T_i^{in} - 1, T_i^{in} - 0.5, T_i^{in}, T_i^{in} + 0.5, T_i^{in} + 1\}$	
Objective	$Maximize\ \sum_{i=0}^{n} \gamma r_i$	
Reward	$r_i = MAX\_COST - c_i,\ c_i(s_i, \alpha_i) = \begin{cases} tr \times E_{norm}(s_i, \alpha_i) + (1 - tr) \times PPD_{norm}(s_i, \alpha_i), & PPD(s_i, \alpha_i) \leq 15 \\ max\_cost, & \text{else} \end{cases}$	
Algorithm	Neural Fitted Q Iteration (NFQ)	
Exploration	Strategy	ϵ-greedy exploration
	Exploration rate (ϵ)	Based on model's prediction (see Eq 12)
Discounting factor - γ	0.98	
Learning Model	Type	Multilayer Perceptron (MLP)
	Activation Function	tanh
	#Layers	4
	#Nodes	16
Validity Definition	Based on	Temporal Difference (TD)
	Criterion for positive decision	$\lvert TD \rvert \leq 0.15$

TABLE II: Summary of building properties.

Building	Surface area	Thermal zones	Operating hours	Warm-up phase	Random occupancy
#1	$350m^2$	8	6:00am–9:00pm	No	Yes
#2	$525m^2$	10	8:00am–9:00pm	Yes	Yes
#3	$420m^2$	10	8:00am–5:00pm	Yes	Yes
#4	$280m^2$	6	7:00am–8:00pm	Yes	Yes
#5	$228m^2$	4	6:00am–6:00pm	No	Yes

TABLE III: Evaluation of yearly results against other methods.

Method	Energy (kWh)	Avg. PPD (%)	Cost (Equation 1)
RBC $20°C$	66,967	24.99	0.89
RBC $21°C$	62,939	17.46	0.72
RBC $22°C$	61,223	11.66	0.59
RBC $23°C$	61,955	7.94	0.52
RBC $24°C$	65,191	6.46	0.51
RBC $25°C$	70,467	7.23	0.56
RBC $26°C$	77,359	10.22	0.66
RBC $27°C$	85,680	15.31	0.81
Fmincon [27]	34,936	6.17	0.33
ESL [25]	36,767	6.54	0.35
Knapsack [28]	36,399	6.47	0.34
Proposed	34,601	7.71	0.36

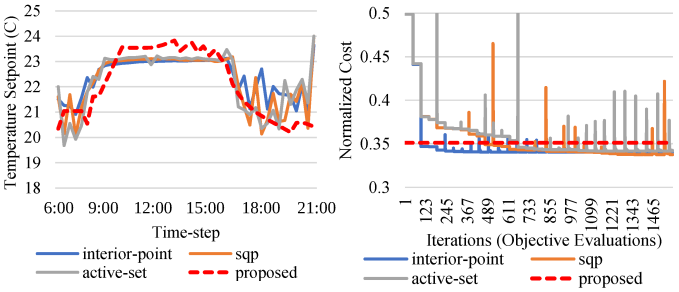
consecutive episodes for an operation period of three months (January–March), while the vertical one gives the duration (in term of operating time-steps) of each episode. Without affecting the generality of our analysis we considered time-steps of 20 minutes duration. Hence, the controller can be involved up to 45 times per day (based on operating hours depicted in Table II). At this figure, an episode with duration 45 indicates that the Main Controller component computes successfully the temperature set-points for the entire day without any failure. Based on our experimentation, the proposed orchestrator achieves an average episode's duration equals to 32 regarding the 3 months (90 days) experiment. In addition to that, if we exclude the training phase during the first 20 episodes, then the corresponding average episode's duration is 37. This figure highlights also that the performance of introduced orchestrator is improved over time, since there is no failure for the last 22 consecutive days. In order to study more thoroughly this efficiency, we also plot at Figure 2 with red color line the outdoor temperature. This analysis indicates that the proposed orchestrator exhibit increased episodes' duration until an unexpected change in weather conditions that has never been encountered before (i.e. at episode 40 the outdoors temperature remains under $15°C$ for the whole day). Similar behavior is reported until the proposed model-free orchestrator to be robust to weather changes (last 22 days).

Apart from episodes' duration, Table III quantifies the efficiency of the proposed orchestrator in terms of improving the overall energy consumption, the average thermal comfort and the total weighted cost. As reference for this comparison are the Ruled Based Configurations (RBCs) ranging from $20°$C up to $27°$C, the well-established *Fmincon* solver [27], as well as the former versions of our decision-making algorithms (the ESL [25] based on support vector machines and the Knapsack optimization algorithm coupled with regression models [28]).

According to this analysis, the proposed orchestrator achieves superior performance against to RBCs, as it achieves overall cost reduction ranging from 41% up to 147%. Moreover, the introduced solution exhibits comparable efficiency against to relevant implementations (e.g. Fmincon, ESL, Knapsack) but without prior knowledge to the employed cost functions and detailed modeling of the target buildings.

In order to study in more detail the efficiency of proposed framework, Figure 3(a) plots the temperature set-points as they are computed with three existing solvers (Interior-Point, SQP and Active-Set) regarding a representative winter day. These methods compute iteratively candidate solutions to solve the problem, where all of them consider an accurate form of the objective functions. On the other hand, the proposed orchestrator does not rely on such an iterative approach (it is executed only once), without considering any information about the employed objective functions. This is also depicted at Figure 3(b), which plots the variation of total cost per iteration for the alternative controllers regarding the same winder day. At this figure, the red dotted line corresponds to the cost of the proposed method.

The efficiency of proposed orchestrator to compute near to optimal results is also quantified at Figure 4, which plots the



(a) Thermostat Set-points (b) Total Cost

Fig. 3: Performance evaluation against standard multi-objective solvers with detailed micro-grid modeling

Pareto results for a representative winder day. The vertical and horizontal axes at this analysis refer to the thermal comfort metric (% PPD) and the buildings' energy consumption, respectively. At this figure three different type of solutions are depicted: (i) the RBC temperature set-points (blue color), (ii) the proposed model-free optimizer (red color) and (iii) the reference Fmincon MPC solver. Each of these methods was invoked with different setups during this analysis. More specifically:

- the RBC selections are in the range 19–25°C with step equals to 0.2°C (30 solutions);
- the relative importance between energy consumption and thermal comfort for the Fmincon solver range from 0 – 1 with step 0.1 (20 solutions);
- the trade-off at the proposed solver ranges between 0 and 1 with step equals to 0.05 (20 solutions).

According to the analysis summarized at this figure, We claim that the proposed method converges to results close to the Pareto front for the majority of executions, which in turn leads to considering the building's thermodynamic behavior and providing close to optimal solutions. Regarding the solutions that deviate from the optimal ones, this occurs due to the random selections during the exploration phase. Finally, we have to state that the solutions retrieved with Fmincon solver (the performance of this solver was already discussed at Figure 3) refer to 10 different trade-off (tr) values.

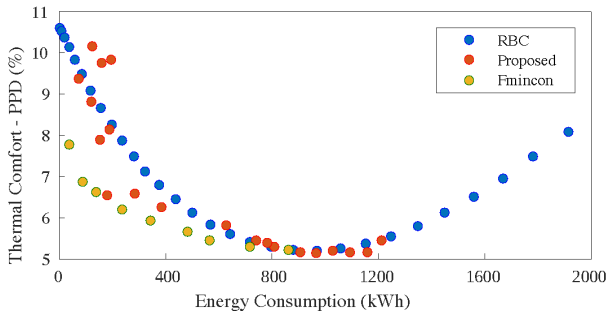


Fig. 4: Pareto analysis results for a representative winter day

The mean TD error based on Equation 13 for the first 90 days is plotted in Figure 5. These results confirm previous

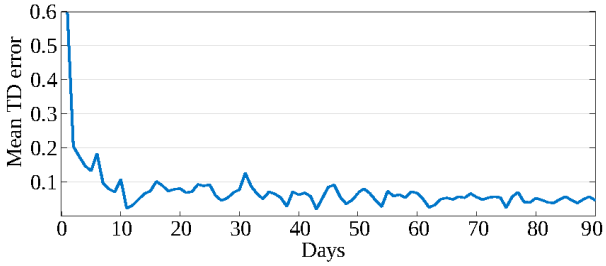
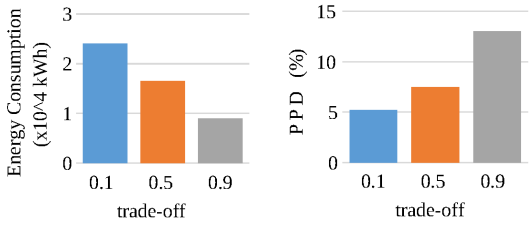


Fig. 5: Evaluation of the Machine Learning model: Daily mean MLP TD-error



(a) Energy consumption (b) Thermal Comfort

Fig. 6: Efficiency of dynamic scaling - the impact of different trade-off values

evidence about improving the controller's efficiency over time, as the machine learning part of the introduced orchestrator leads to lower error values. More specifically, the majority of these values is less than 0.15 (average error value is 0.07), which has been defined as the threshold value for considering the model as successful.

As discussed in Section III, the optimization objective is formulated as a weighted sum, which enables to define the relative importance of energy consumption and occupants' thermal comfort metric. However, in order to fully support this relative importance, our framework incorporates also a dynamic scaling kernel. At the experiment illustrated in Figure 6 we study this feature for alternative weight factors, referred to as trade-off weights. In detail, $tr = 0.1$ corresponds to the maximum occupant's thermal comfort, while $tr = 0.9$ leads to the maximal energy savings without exceeding the threshold of 15% for the thermal comfort threshold. Based on these results, we observe that our system respects the designer intention by adapting the importance of each objective according to tr values.

B. Evaluate Orchestrator's Run-Time Overhead

The task of model optimization is performance dominant. Hence, the execution time of this component is the crucial one that defines the overall orchestrator's performance. In order to study the proposed framework's run-time overhead, it was implemented onto a low-cost embedded system. The target platform for this analysis is a 4-core ARM Cortex A57 operating at 1,900 MHz with 4 GB of system memory. Figure 7 plots the execution latency for representative input data sets

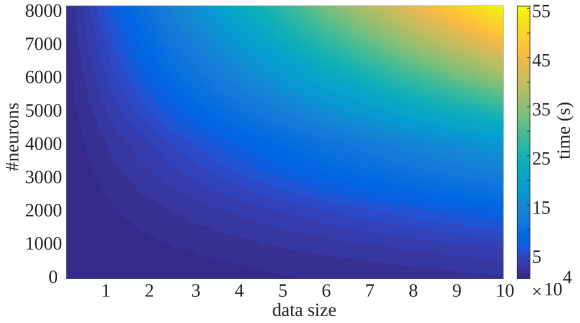


Fig. 7: Run-time for model optimization (data transfers, building targets, 1 epoch re-training) on ARM Cortex A57.

and number of neurons per ANN. The ANN architecture for this experiment consists of 2 up to 18 layers. Each of the solution has an average of 5 input features that correspond to environmental variables acquired from building's sensors.

The functionality of the proposed orchestrator discussed throughout this manuscript incorporates an MLP with 4 input nodes (corresponding to the 4 features of the state), 2 hidden layers with 16 nodes and a data window size less than 3,000 points. Based on our experimentation, the average delay per time-step for this orchestrator is 0.03 seconds, which is enough for the decision-making task imposed by a typical HVAC configuration scenario (i.e. 20 minutes time frame). More specifically, as it was already depicted at Figure 2, there were only 55 re-trainings for the first 3 months of operation. Consequently, the time delay is acceptable, since the training phase can be executed in parallel with the continuous operation. Even for the border case, where a full retraining is necessary, then the employed model causes an overhead of less than 1 second for the ARM Cortex A57.

V. Conclusions

A framework that supports the design of a low-cost CPS orchestrator targeting HVAC systems, was introduced. This orchestratos was applied in order to optimize a multi-objective problem related to the simultaneous enhancement of buildings' energy consumption and the occupant's thermal comfort metric. Based on our experimentation, we validate the superiority of proposed solution against state-of-the-art relevant solvers without the necessity of accurate prior system modeling, as both functions that describe energy consumption and thermal comfort are agnostic. Also, the introduced solution exhibits significant lower computational/storage complexities, which in turn enables its execution onto low-cost embedded devices.

References

[1] Eurostat, "Energy balance sheets," Data 2002–-2003, Luxemburg, 2005.

[2] F. D. Macías-Escrivá and et.al, "Self-adaptive systems: A survey of current approaches, research challenges and applications," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7267 – 7279, 2013.

[3] T. Baúar and P. Bernard, "Optimal control and related minimax design problems," 1995.

[4] W.-M. Lu and J. Doyle, "$h\infty$ control of nonlinear systems: a convex characterization," *Automatic Control, IEEE Transactions on*, vol. 40, no. 9, pp. 1668–1675, Sep 1995.

[5] C. D. Korkas, S. Baldi, I. Michailidis, and E. B. Kosmatopoulos, "Intelligent energy and thermal comfort management in grid-connected microgrids with heterogeneous occupancy schedule," *Applied Energy*, vol. 149, pp. 194 – 203, 2015.

[6] A. Afram and F. Janabi-Sharifi, "Theory and applications of hvac control systems–a review of model predictive control (mpc)," *Building and Environment*, vol. 72, pp. 343–355, 2014.

[7] J. Clarke and et.al., "The role of simulation in support of internet-based energy services," *Energy and Buildings*, vol. 36, no. 8, pp. 837 – 846, 2004.

[8] L. Magni, G. De Nicolao, L. Magnani, and R. Scattolini, "A stabilizing model-based predictive control algorithm for nonlinear systems," *Automatica*, vol. 37, no. 9, pp. 1351–1362, 2001.

[9] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[10] A. I. Dounis and C. Caraiscos, "Advanced control systems engineering for energy and comfort management in a building environment—a review," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 6-7, pp. 1246–1261, 2009.

[11] E. Barrett and S. Linder, "Autonomous hvac control, a reinforcement learning approach," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2015, pp. 3–19.

[12] T. Wei, Y. Wang, and Q. Zhu, "Deep reinforcement learning for building hvac control," in *DAC*. IEEE, 2017, pp. 1–6.

[13] Y. Chen, L. K. Norford, H. W. Samuelson, and A. Malkawi, "Optimal control of hvac and window systems for natural ventilation through reinforcement learning," *Energy and Buildings*, vol. 169, pp. 195–205, 2018.

[14] K. Dalamagkidis and D. Kolokotsa, "Reinforcement learning for building environmental control," *Reinforcement Learning, Theory and Applications. I-Tech Education and Publishing*, pp. 283–294, 2008.

[15] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Structural and multidisciplinary optimization*, vol. 41, no. 6, pp. 853–862, 2010.

[16] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1.

[17] M. Riedmiller, M. Montemerlo, and H. Dahlkamp, "Learning to drive a real car in 20 minutes," in *Frontiers in the Convergence of Bioscience and Information Technologies*. IEEE, 2007, pp. 645–650.

[18] D. Bollegala, "Dynamic feature scaling for online learning of binary classifiers," *Knowledge-Based Systems*, vol. 129, pp. 97–105, 2017.

[19] T.-H. Teng, A.-H. Tan, and Y.-S. Tan, "Self-regulating action exploration in reinforcement learning," *Procedia Computer Science*, vol. 13, pp. 18–30, 2012.

[20] C. Gehring and D. Precup, "Smart exploration in reinforcement learning using absolute temporal difference errors," in *Int. Conf. on Autonomous agents and multi-agent systems*, 2013, pp. 1037–1044.

[21] U. S. Department of Energy, "Energyplus energy simulation software," http://apps1.eere.energy.gov/buildings/energyplus/, 2015.

[22] D. B. Crawley, L. K. Lawrie, F. C. Winkelmann, W. Buhl, Y. Huang, C. O. Pedersen, R. K. Strand, R. J. Liesen, D. E. Fisher, M. J. Witte, and J. Glazer, "Energyplus: creating a new-generation building energy simulation program," *Energy and Buildings*, vol. 33, no. 4, pp. 319 – 331, 2001.

[23] "Wholesale electricity and natural gas market data," November 2015. [Online]. Available: http://www.eia.gov/electricity/wholesale/#history

[24] Weather Data, "Weather data sources for energyplus framework," 2015. [Online]. Available: http://apps1.eere.energy.gov/buildings/energyplus/weatherdata_sources.cfm/

[25] C. Marantos, K. Siozios, and D. Soudris, "A flexible decision-making mechanism targeting smart thermostats," *IEEE Embedded Systems Letters*, vol. 9, no. 4, pp. 105–108, 2017.

[26] M. Riedmiller, "Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method," in *European Conference on Machine Learning*. Springer, 2005, pp. 317–328.

[27] R. H. Byrd, J. C. Gilbert, and J. Nocedal, "A trust region method based on interior point techniques for nonlinear programming," *Mathematical Programming*, vol. 89, no. 1, pp. 149–185, 2000.

[28] C. Marantos, K. Siozios, and D. Soudris, "Rapid prototyping of low-complexity orchestrator targeting cyberphysical systems: The smart-thermostat usecase," *IEEE Transactions on Control Systems Technology*, 2019.