A Guide for the Design of Benchmark Environments for Building Energy Optimization

David Wölfle
FZI Research Center for Information Technology
woelfle@fzi.de

Arun Vishwanath
IBM Research Australia
arvishwa@au1.ibm.com

Hartmut Schmeck
Karlsruhe Institute of Technology
hartmut.schmeck@kit.edu

ABSTRACT

The need for algorithms that optimize building energy consumption is usually motivated with the high energy consumption of buildings on a global scale. However, the current practice for evaluating the performance of such algorithms does not reflect this goal, as in most cases the performance is reported for one specific simulated building only, which provides no indication about the generalization of the score on other buildings. One approach to overcome this severe issue is to establish a shared collection of environments, each representing one simulated building setup, that would enable researchers to systematically compare and contrast the efficacy of their building optimization algorithms at scale. However, this requires that the individual environments are well designed for this goal. This paper is thus targeting the design of suitable environments for such a collection based on a detailed analysis of related publications that allows the identification of relevant characteristics for suitable environments. Based on this analysis a guide is developed that distills these characteristics into questions, intended to support a discussion of relevant topics during the design of such environments. Additional explanations and examples are provided for each question to make the guide more comprehensible. Finally, it is demonstrated how the guide can be applied, by utilizing it for the design of a novel environment, which represents an office building in tropical climate. This environment is released open source alongside this publication. We also indicate how test scenarios from existing publications could be enhanced to comply with the required characteristics according to our guide, underlining its importance for the future development and evaluation of building energy optimization algorithms, and thus for the sustainability of buildings in general.

CCS CONCEPTS

• **Software and its engineering** → **Designing software**; **Development frameworks and environments**.

KEYWORDS

building energy management, building control, environment, benchmark, evaluation, reinforcement learning, smart building

BuildSys '20, November 18–20, 2020, Virtual Event, Japan

ACM ISBN 978-1-4503-8061-4/20/11.
https://doi.org/10.1145/3408308.3427614

ACM Reference Format:
David Wölfle, Arun Vishwanath, and Hartmut Schmeck. 2020. A Guide for the Design of Benchmark Environments for Building Energy Optimization. In *The 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '20), November 18–20, 2020, Virtual Event, Japan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3408308.3427614

1 INTRODUCTION

An enormous amount of scientific work has been published within the last decades covering the algorithmic optimization of energy consumption patterns of buildings, usually with the target to increase energy efficiency, provide flexibility, and/or improve user comfort [23]. However, while it is very common to motivate the research by referring to the high share of *global* energy consumption caused by buildings, there is a tendency to evaluate the proposed algorithms very *locally*, in many cases even on exactly one building that is furthermore often closely related to the institution the authors are affiliated with (see e.g. [1, 6–8, 16, 17, 20, 21, 32]). Obviously, this contradicts the original intention of developing algorithms that address a global scale problem, i.e. that can be applied to a larger number of potentially very diverse buildings. However, from all publications addressing the algorithmic optimization of buildings reviewed for this work [1, 6–8, 12, 16–18, 20, 21, 30, 32, 33] none did actually discuss the impact of the evaluation environment on the results. Furthermore, the observed single building evaluation procedure does not only hinder the interpretation of the actual performance of algorithms, it also makes comparisons between publications nearly infeasible [31]. This is due to the evaluation of algorithms in non-standardized settings and to the strong dependence of the reported scores on the specific evaluation environment which prevents direct comparisons.

To mitigate these issues it has been proposed to establish a collection of shared environments, i.e. a number of open source programs, as a common foundation for the objective and systematic evaluation of building energy optimization algorithms [31]. In this context, each environment represents a test case, i.e. a typical scenario that could be the target for such an optimization. An example could be the optimization of the self-consumption rate of photovoltaic panels in combination with a battery storage system or, alternatively, the optimization of the energy costs of an air conditioning system. While each environment alone may already serve as a valuable tool, the essential idea is that a collection of environments should reflect the diversity of buildings in general, thus effectively allowing the systematic and objective evaluation of building energy optimization algorithms in a global perspective. However, such a collection of benchmark environments will certainly only be of use

if the individual members are "good" environments for the purpose of evaluating building energy optimization algorithms. Thus, the natural question arises, what defines a "good" environment, and how one could achieve the creation of such a "good" environment.

Hence, this paper contributes a guide for the systematic design of benchmark environments to evaluate building energy optimization algorithms. Effectively, we provide a collection of topics and questions that one should consider or discuss while designing such an environment. In order to ensure that our proposed guide is as complete as possible, we carried out an extensive analysis of topics that are currently perceived important for design and application of benchmark environments in the context of building energy optimization. The results of this analysis are documented in Section 3. The actual guide is presented in Section 4. While it appears generally sufficient to verbally discuss the introduced topics and guiding questions while designing an environment, the main intention is to document the outcome in a scientific publication in order to allow others to comprehend the reasoning behind this design. In order to provide a practical example for this process, we use our guide to design a new benchmark environment, and document this in Section 5. Furthermore, we discuss how a few more known scenarios could be enhanced to become environments complying with the criteria of our guide. Finally, we summarize our work in Section 6. Since the process of evaluating optimization algorithms is subject to many different scientific fields, and in order to prevent misinterpretation, we begin in Section 2 by briefly introducing a nomenclature that is used throughout our paper.

2 NOMENCLATURE

A *benchmark* is commonly understood as the actual performance evaluation of optimization algorithms [3, 9, 11] on a selected test set. Carrying out a benchmark is a non-trivial task, a best practice guide to do so is e.g. given by Beiranvand et al. [3], but for this paper it is sufficient to perceive a benchmark as a process of:

- Selecting a set of optimization algorithms.
- Selecting tasks used for evaluating the algorithms.
- Executing each task with each optimization algorithm and reporting the obtained performance.

Following common convention [4, 9, 13, 25], we will denote a task within a benchmark as *environment*. The environment operates as a sequential decision process with discrete time stepping. At each time step, the environment yields a set of *observations* and *rewards* as output. The former reflects the measured sensor values while the latter usually corresponds to the quantity being optimized. The optimization algorithm is expected to compute a set of *actions*, these are the decision variable values, that are passed to the environment. Based on these, the environment will advance one time step and emit a new set of observations and rewards. This is iterated until eventually a terminal state is reached. A practical example could be the air conditioning of a room, for which the observations could include the current temperature and the desired setpoint. The reward signal could reflect the energy costs, usually in addition to a penalty term for violating comfort constraints, while the actions could correspond to the opening level of a valve controlling the exchange of cool water between the air handling unit and the heat exchanger. Finally, the environment denotes the room with all hardware, or relevant to our work, a simulation code of the same.

It should be noted that some authors tend to use the term *test function* instead of environment. This is in particular the case for publications addressing optimization algorithms that are not tailored to specific applications, like e.g. general black box optimization [11]. However, in such scenarios the test tasks are usually actual functions, in a mathematical sense, while the term environment rather expresses the use of complex simulation code as we expect it for the building energy optimization case addressed in this paper. We will thus stick to the term environment by default, apart from situations in which we explicitly like to express the meaning above which is a mathematical function.

It should also be noted that the interaction with environments in terms of actions, observations, and rewards at every time step is heavily influenced by the conventions of reinforcement learning research. However, it has been argued that the concept is general enough to allow the interaction with other types of optimization algoritms [31], which is why it has been applied in this work.

3 STATE OF THE ART

In order to ensure that our proposed guide is as complete as possible, an extensive analysis to identify characteristics of "good" benchmarks for the evaluation of building energy optimization algorithms was carried out. However, to our best knowledge and research efforts, we could not identify a single scientific publication that covered this topic explicitly. We thus focused on analyzing closely related work, which is introduced in the following subsections. While reviewing these publications, it was not possible to identify hard measures for "good" environments. Instead, we identified shared topics, like e.g. *scenario* or *performance measure*, that reappear in different publications and that thus seem to be important for the environments. Therefore, this section introduces these as they appear in the reviewed literature. For the development of the guide in Section 4 we discuss and elaborate on the relevance of these topics in greater detail.

3.1 Publications Related to Building Energy Optimization Benchmarks

The "CityLearn v1.0" environment [26] is by far the closest related work found. The goal is to control the heat pumps and batteries of ten buildings in parallel, such that these reduce energy consumption and minimize peaks [27]. However, neither the corresponding publication [26] nor the project website [27] provide any discussion about the design of it, apart from motivating the necessity for demand response and potential benefits of applying reinforcement learning algorithms to control buildings. The environment is in fact heavily inspired by OpenAI Gym, which is discussed in Subsection 3.4, including implications for the design of environments for the evaluation of building energy optimization algorithms.

In [29] Waibel et al. discuss the suitability of applying mathematical test functions to evaluate algorithms for building energy optimization. They argue that it is relevant to examine the *scope* of the optimization problem, here especially the fitness landscape, and apply a method called fitness landscape analysis to compare 15 EnergyPlus simulations with 24 mathematical test functions.

They also found it necessary to discuss the *relevance* of their selected buildings, for elaborating the representativeness of these for buildings in general.

3.2 Publications Related to Building Energy Optimization Algorithms

A large amount of publications is available covering algorithms for optimized operation of buildings and their hardware components in order to minimize energy costs and consumption, maximize user comfort, and/or reach a desired energy consumption pattern. Usually, for showing the relevance of the introduced optimization algorithm, the authors tend to select and utilize a benchmarking environment for demonstrating the desired performance. In order to assess the current practice for the design and application of benchmark environments we began by directly searching for relevant publications, from which we have chosen a representative set [17, 18, 20, 21] for explicit reference. To broaden our focus beyond the sorting logic of typical search engines, we also considered [1, 8, 12, 16, 30], which have been arbitrarily selected from [23]. The latter is an extensive review about optimized control systems for building energy and comfort management. Finally, scanning the proceedings of the ACM BuildSys 2018 and 2019 yielded [6, 7, 32, 33], which represent more recent developments. The remainder of this subsection will summarize the findings, i.e. how benchmarking environments are selected and utilized in these publications.

As a first finding it can be noted that none of the investigated publications [1, 6–8, 12, 16–18, 20, 21, 30, 32, 33] did use an environment that was explicitly designed to objectively compare building energy optimization algorithms, most likely as such environments do not exist yet. The *scope* of the utilized environments is therefore rather closely tailored towards the evaluated algorithms, i.e. the environments seem to be explicitly engineered to match the optimization algorithm. An example is the formulation of the environment as a linear program to evaluate the performance of a linear optimization algorithm [18].

On the other hand, it seems common convention, e.g. in [1, 6, 7, 12, 17, 18, 20, 21, 32, 33], to describe the *scenario* represented by the environment. Such elaboration usually introduces the optimization target but also covers information about the represented building itself, as well as the considered devices relevant for building energy management. The provided level of detail varies between very high level information like location and floor area, e.g. in [7, 8, 12], to fine grained description of the construction, like heat loss coefficients of facade elements in [21]. Furthermore, it is also common to describe the way the scenario has been modeled. The model is either described as a set of mathematical equations, e.g. in [1, 16, 18, 20], or it is made clear that the scenario has been implemented in a building simulation software, often EnergyPlus as in [6, 7, 17, 21, 32, 33].

Finally, all of the investigated publications implicitly or explicitly defined a *performance measure*, i.e. one ore more metrics that indicate how well the algorithm performs w.r.t. the environment.

3.3 Publications Related to Benchmarking Optimization Algorithms

Similar to the large corpus covering optimization algorithms for building energy optimization, there also seems to exist a wide range of scientific work that carried out benchmarks for optimization algorithms in general. For example, Beiranvand et al. [3] list over 50 of such publications, which we will not review here as it appears out of scope. However, the work of Beiranvand et al. itself is absolutely relevant for this work as they discuss best practices for executing benchmarks for optimization algorithms. Although they do not address the design of test functions or environments explicitly, some implications for "good" design can be found.

The first point of these implications targets the *relevance* of the environment, as they propose to clearly define the reason of the benchmark before it is carried out. Furthermore, they stress that the environments utilized in the benchmark must be relevant in that sense, i.e. the environments must be capable to evaluate the optimization algorithms w.r.t. the investigated reason. A second implication aims at the *realism* of the environment, as they suggest that the environment should reflect a real world problem (here in contrast to a designed test function), if the benchmark aims to evaluate the performance of optimization algorithms for a practical application. Going on, a larger fraction of [3] is dedicated to the design of a *performance measure* to evaluate the optimization algorithms based on the collection of environments used in a benchmark. The suggested measures include the typical choices like accuracy and number of constraint violations, but also running time and memory usage. Furthermore, Beiranvand et al. suggest to prefer environments with known optimal solutions, as this allows a more objective evaluation of the performance. Finally, they address the *scope* by stating that it is common to group test functions by mathematical types, e.g. whether the optimization problem is convex or not.

The latter point is confirmed by Hansen et al. [11], which introduce a collection of test functions for the evaluation of black-box optimization algorithms. Furthermore, they define what information is gained for each test function if one uses this particular function in a benchmark. For example, they state that a simple spherical function yields information about the optimal convergence rates, while most other test functions they propose are not convex and thus more difficult.

3.4 Publications Related to Environments for Reinforcement Learning

Reinforcement Learning (RL) can generally be perceived as an approach that aims to learn optimal sequential decisions from interactions with an environment [25]. Although research on RL has been carried out for several decades, its popularity has risen significantly within recent years, which also led to many applications in building energy management, e.g. by [6, 7, 32, 33]. The rising popularity is very likely due to outstanding results in algorithmic game playing, e.g. on Atari [15] or Go [24]. However, it appears that these important advances were significantly facilitated by the consequent separation of the development of environments and algorithms, similar to our approach. In fact, the fundamental ideas of this work are inspired by the development in RL in recent years.

The general concept of designing test problems to evaluate the performance of RL algorithms is certainly as old as RL itself. However, the concept of developing designated environments, in a sense used in this work, i.e. as software components conceived for the

objective comparison of algorithms, is much younger. One of the pioneers of such environments is certainly the Arcade Learning Environment (ALE) [4] that is intended to evaluate the performance of RL agents in general game playing, and that has been applied in many high impact publications in the last years, e.g. [14, 15, 22]. Interestingly, the authors of the ALE provide no discussion about its suitability to evaluate RL agents, apart from stating that the included games are a challenging problem for RL. More relevant here is the work by Machado et al. [13] that identifies conceptual design flaws of the ALE by summarizing five years of application by the scientific community. One of the most severe issues is related to the computation of the *performance measure*, or more precisely, the absence of a definition how the performance measure should be computed. This has led to different definitions of the performance measure by different authors applying the ALE, thus leading to poor comparability of the results. A second issue is related to the *reproducibility*. While it is immediately obvious that the environment should yield reproducible results, this has not always been the case for the ALE. One reason for this has been poor clarity about the intended usage pattern, e.g. how long a game is played before a score is reported. Additionally, software updates to the ALE's source code have been identified to reduce the comparability of results. As a result of the latter, it is now common to report environments with name and a version number, e.g. in [26].

The rising popularity of environments like ALE has brought the usability of these environments more into focus. As a consequence, software collections have emerged that bundle several already available environments and allow the interaction with these environments over a standardized *interface*. The most popular of these projects is OpenAI's Gym [5, 19] which established the current de facto standard for interaction between algorithm and environment.

4 A GUIDE FOR THE DESIGN OF SYSTEMATIC BENCHMARK ENVIRONMENTS

The analysis of the state of the art in the section above has yielded a diverse set of characteristics that seem important for "good" environments. Furthermore, it was possible to identify common topics, that can be used to cluster the characteristics into groups. Based on these findings this section presents our main contribution, that is a guide for designing systematic benchmark environments for the evaluation of building energy optimization algorithms. To this end we distilled the most important characteristics of "good" environments into guiding questions, i.e. questions that we would expect authors of environments to answer within a corresponding publication or documentation. These guiding questions are grouped according to the associated topics, i.e. each of the following subsections is devoted to one topic. Each subsection begins with a definition of the topic, followed by the corresponding guiding questions and an explanatory text which specifies the question in more detail, but also contains reasoning why the particular question is important. Usually, we also present a short example.

4.1 Scenario

Considering the results presented in Subsection 3.2, the most fundamental topic for an environment is certainly the scenario, as it defines what is actually being represented by it.

What is the optimization target?
Following common convention outlined in Subsection 3.2 it is certainly necessary to define the optimization target while describing the scenario. The goal is to explicitly clarify the target for any optimization algorithm evaluated with the proposed environment. Typical examples could be the minimization of energy consumption or the energy costs. Any constraint that must be considered during optimization, e.g. temperature ranges to maintain thermal comfort, should be listed too.

What characterizes the represented building and its devices?
Similar to the previous guiding question it is also common convention to define the building represented by the environment as well as the devices within the building, at least those that are relevant for building energy management. The description should be sufficiently detailed to allow discussion about the adequacy of the model to reflect the building, but also provide the necessary information for debating the relevance of the environment. Examples for potentially relevant characteristics of buildings and devices are given in Section 3.2.

How is the building/are the devices modeled?
Specific information about how the building along with the devices is modeled is certainly necessary as a foundation for discussing other topics, like scope or realism, and should be provided at appropriate level of detail accordingly, e.g. by listing relevant equations. Additional information, however, is certainly better handled by referring to source code published alongside, especially for very complex models like e.g. those used by the EnergyPlus software.

How is stochasticity dealt with?
Stochasticity is naturally part of building energy optimization, e.g. as measurement error or when weather forecasts are incorporated into the optimization process [18]. Furthermore, stochastic elements have been found to make environments more challenging and realistic [13]. We therefore suggest to systematically analyze the scenario for stochastic elements, and model these in the environment accordingly.

4.2 Relevance

As highlighted in Subsection 3.3, it is certainly important to clarify the reason for benchmarking a priori. In the context of this work such reason is clearly to evaluate the general performance of optimization algorithms for building energy management. What is left to consider during the design of an environment is thus the relevance of the environment for this particular application, i.e. why a user should use the proposed environment.

How representative is the scenario for buildings in general?
Following our findings in Subsection 3.1 and 3.2, a first point for debating the relevance of the environment is certainly the expressiveness of the computed metrics for buildings in general. One should therefore define which buildings are represented by the environment. As an example, one might show that the air conditioning system represented in a proposed environment is typical for office buildings in Northern America.

How is the environment different from existing environments?
Discussing the differences of the proposed environment from existing ones should support users while choosing environments for a

particular benchmark. For example, it could be pointed out that the environment targets office buildings while existing environments represent private households.

4.3 Scope

During our analysis of the state of the art in Section 3 it has been found that most environments are implicitly or explicitly designed for the evaluation of certain types of optimization algorithms. We refer to this as *scope*.

What types of optimization algorithms are addressed?
Any publication introducing an environment should define the scope to allow potential users to evaluate if the environment is suitable for the candidate optimization algorithms. Defining the scope is straight forward if the environment has been designed for the evaluation of certain types of optimization algorithms, by simply naming the targeted algorithms. An example is the CityLearn challenge introduced in Subsection 3.1 that explicitly addresses reinforcement learning algorithms. However, environments appear more valuable if they can be applied to wider ranges of algorithms, i.e. have a broader scope. We thus recommend to not design environments for specific types of algorithms.

What characterizes the underlying optimization problem?
The evaluation whether a specific algorithm is able to solve an environment is certainly harder if the environment does not address certain types of algorithms. Following the findings listed in Subsection 3.3 one should therefore characterize the underlying optimization problem to support such evaluations. One might for example specify that the optimization problem underlying a proposed environment is non-linear but convex.

4.4 Realism

As pointed out in Subsection 3.3 it is good practice to evaluate optimization algorithms that are developed for certain applications in environments that reflect these applications realistically. In our application we consider an environment to be realistic if there is no difference for an optimization algorithm between interacting with the environment or the building represented by it.

Are the sensors and actuators typically available in the represented scenario?
As a first point to discuss the realism we propose to question the assumed hardware configuration for the represented building. It is for example often assumed (e.g. in [6, 7, 18]) that detailed occupancy information is available and can be used for optimization. During the design of the environment one might thus discuss, if occupancy sensors are typically available in the represented buildings, and if not, if retrofitting a presence detection system would be economically viable. One might furthermore consider if the data of the proposed hardware is available for optimization, as e.g. legal or privacy concerns need to be considered.

How does the runtime of the optimization algorithm affect the realism of the environment?
A second point addresses that some optimization algorithms tend to require longer periods (minutes/hours) to compute an output. On the other hand common environments, e.g. [4, 19], do simply pause the simulation while awaiting input from the optimization algorithm. This becomes certainly unrealistic if the runtime of the optimizer is not much smaller than the time step length of the environment, and should thus be considered during design.

How is the realism of the environment affected by modeling simplifications?
Naturally, any simulation model of reality is simplified to some extent. An example for such a simplification in the context of this work is the zone temperature in EnergyPlus simulation, which is assumed to be constant everywhere in the zone. While such simplifications may have no practical impact on the realism of the environment, this point should nevertheless be discussed.

4.5 Performance Measure

Following the concepts introduced in Subsections 3.2 to 3.4 the performance measure refers to one or more numeric values that summarize the performance of the evaluated optimization algorithm w.r.t. a specific environment and that is used to compare the performance of several algorithms with each other.

How is the performance measure computed?
Defining performance measures traditionally does not seem to be part of the environment design, but is rather left to the person carrying out a benchmark [3, 4]. However, this procedure leads to poor comparability between reported scores using the same environment, which is considered a severe issue [13]. We therefore suggest to define the performance measure alongside the environment including discussion why the performance measure is relevant for evaluating and comparing optimization algorithms. As an example one might use the relative energy cost savings as performance measure for an environment reflecting a battery storage system in a private household, and motivate this decision by stating that the monetary savings potential is likely the main motivation for purchasing a system that optimizes the operation of the storage system.

Which values of performance measure can be considered a bad or good result?
Considering the publications listed in Subsection 3.2, it appears common to report performance measures for optimization algorithms targeting buildings as an improvement compared to a baseline controller. An example for such a procedure is given by Chen et al. [6] who reported energy savings of 16.7% for an air conditioning system compared to the currently installed controller. However, without any additional details it is hard to interpret whether this performance is actually good or not. To overcome this issue we suggest to discuss which values of the performance measure can be considered a bad or good result. In order to identify bad performance we suggest to keep the current procedure, i.e. to provide the performance measure of a baseline. However, it appears especially relevant that the baseline is a strong one and reflects the current state of the art. An example is given by Ding et al. [7] who extensively discuss the quality of the controller that they use for comparison. In order to additionally identify good performance it is commonly suggested, e.g. in [3, 11], to provide the performance measure of the optimal solution for comparison. If the optimal solution is unknown and cannot be estimated, it appears reasonable to define a threshold that allows the identification of a good solution. An example for such a procedure is the application of reinforcement

Table 1: Overview of relevant topics for environments.

Topic	Description
Scenario	Defines what is represented by the environment, the relevant parts of the building, the modeling of these, and the optimization target.
Relevance	Describes why one should use the environment, the representativity of the environments for buildings in general, and differences from existing environments.
Scope	Designates whether the environment is designed for specific optimization algorithms and characterizes the underlying optimization problem.
Realism	Discusses the realism of the environment considering the assumed hardware, runtime requirements, and modeling simplifications.
Performance Measure	Delineates how the score that is emitted by the environment is computed and provides details to distinguish good from bad results.
Reproducibility	Depicts how the environment must be used to reproduce the results and how the environment should be referred to.
Interface	Determines how to exchange information with the environment including the format and ranges of the data.

learning algorithms to computer games, where the highest possible score is often unknown and a strong human player is chosen as reference.

4.6 Reproducibility

As highlighted in Subsection 3.4 it is important that environments deliver reproducible results, i.e. experiments can be repeated and verified by others yielding identical performance measures.

What instructions must be followed to utilize the environment in its intended way?
Following the findings of [13], we suggest to provide clear usage instructions as a first step towards reproducibility. Besides describing how a user should interact with the interface of the environment (see Subsection 4.7) these instructions should also cover how the environment must be installed to ensure the intended operation. If the environment contains stochastic elements that affect the reported performance metric, the user should also be advised how to behave correctly in that light. For example, it is common for reinforcement learning publications to repeat the interaction with the environment five times and report the average score.

How should the environment be referred to?
As a second step towards reproducibility we suggest to define a unique and novel name for the environment. Considering the results outlined in Subsection 3.4, we furthermore encourage the usage of a version number, to prevent inconsistency if breaking changes must be made to the source code of the environment.

4.7 Interface

In order to evaluate a building energy optimization algorithm with an environment, it is necessary to connect both programs. Here we refer to the part of the environment that is exposed to the tested algorithm as interface. Thus the interface allows the transportation of observations and rewards from the environment to the optimization algorithm, which is expected to compute a new action, which is forwarded back to the environment with an interface call.

Which conventions does the interface follow?
While one could freely choose the design of the interface while developing an environment, our findings of Subsection 3.4 have indicated that it is instead common to follow common conventions. This procedure is likely popular as it allows faster implementation of both, the environment and the tested algorithm, the latter especially if many environments should be used for benchmarking. An example for such a convention is OpenAI's Gym.

What is represented by the values of observations, rewards and actions?
It is common practice for interfaces to just exchange the bare data and define how this data should be interpreted in some documentation that is provided alongside. Interfaces of environments, e.g. in [6, 17, 27, 32], typically proceed similarly, that is, they provide observations, actions, and rewards as bare numbers or arrays of such, while providing additional information on how the data should be interpreted in the publication. If the environment would reflect the air conditioning system of a room, for example, one would define that the observations are provided as an array of two floats (e.g. $[22.8, 49.0]$), the first representing the current temperature and the second the current humidity in the room. Rewards and actions should be defined accordingly.

What is the allowed value range for observations, rewards and actions?
In addition to the previous guiding questions it is also good practice, e.g. in [17, 29], to define the possible value range for observations, rewards, and actions, as the range is usually relevant for adapting algorithms to the application. Extending the example above one would thus define that the possible values for the observations are $\{5, \ldots, 40\}$ for temperature and $\{0, \ldots, 100\}$ for humidity.

4.8 Summary

A brief overview of relevant topics that should be considered during the design of environments for the evaluation of building energy optimization algorithms is presented in Table 1.

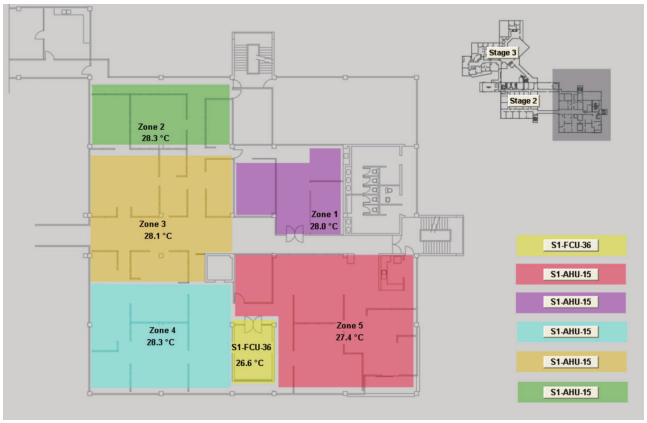

Figure 1: Schematic of floor layout and the thermal zones. Climate control to this section is provided by one AHU.

5 APPLICATION OF THE GUIDE

In order to demonstrate the relevance of the guide introduced above, this section is devoted to its practical application. Our approach is twofold. In the first subsection, we provide an example for the design of an environment utilizing the guide as it was intended, i.e. by providing answers and discussion w.r.t. the guiding questions. This allows us to demonstrate that the guide is a useful tool for the design of environments, while also providing a complete example for a scientific documentation of such a design process. Furthermore, it also allows us to publish a ready-to-use environment alongside this publication, a link is provided in Supplementary Material. The second subsection demonstrates the general utility of the guide by discussing potential benefits of utilizing it for enhancing test environments from other publications.

5.1 The TropicalPrecooling Environment

This subsection introduces the TropicalPrecooling environment in detail. In order to prevent the description from becoming too voluminous or even out of scope for this publication, it was decided to build up on a scenario published in [28] that is the reduction of energy costs incurred by a heating, ventilation, and air conditioning (HVAC) system to cool a commercial building in tropical climate, while ensuring thermal comfort to the occupants. Parts of the following content, especially the scenario description, have thus already been published in [28]. Here, we provide sufficient detail, for clarity of exposition in the context of this paper, in order to give a good example for documenting an environment. Furthermore, it should be noted that the original publication primarily focused on the evaluation of one specific optimization algorithm, while in this publication we address the development of a benchmark environment.

5.1.1 Scenario. The building considered in this environment is an office and administration building located in a warm tropical city in the southern hemisphere. In this location, no heating is required in the winter months. The building houses about 250 people. The office hours are between 8 am and 5 pm Monday to Friday. The rates structure contracted with the local electricity utility is as follows. There is a daily service fee of $2.68, a peak consumption charge of 24.48 cents/kWh when electricity is consumed between 7 am and 11 pm and an off-peak consumption charge of 9.78 cents/kWh for electricity consumed outside of these hours. The lower and upper zone temperature comfort bounds, as specified by the facility manager, are given by:

$$T^{max}_{comfort}(t) = \begin{cases} 25\ ^{\circ}C\ ; \text{if } t \in \{7\text{ am}, \ldots, 5\text{ pm}\} \\ 29\ ^{\circ}C\ ; \text{otherwise} \end{cases} \quad (1)$$

$$T^{min}_{comfort}(t) = \begin{cases} 23\ ^{\circ}C\ ; \text{if } t \in \{7\text{ am}, \ldots, 5\text{ pm}\} \\ 25\ ^{\circ}C\ ; \text{otherwise} \end{cases} \quad (2)$$

The time window {7 am, . . . , 5 pm} is the core occupancy hours, meaning it corresponds to the continuous time period when the building and the zones are mostly occupied. However, this is only applicable on weekdays as the HVAC system is turned off by default on weekends and after 5 pm on weekdays, unless people are working late. Therefore, the optimization target is the minimization of energy costs under consideration of the thermal comfort constraints, i.e. exploiting the off-peak consumption rates in the morning by precooling the building before the usual occupancy hours. Following the current operation strategy of the building, the optimization algorithm is requested to compute a schedule of setpoint temperatures for the time window {4 am, . . . , 5 pm} once per weekday at 2 am, in order to exploit the precooling potential.

The building consists of several sections that can be controlled independently and the optimization challenge is similar for all those. It is thus sufficient to model only one section to represent the challenge of optimizing the building in total. The environment reflects therefore Stage 1 Level 1, which is comprised of five thermal zones, as shown in Fig. 1. Cooling to the building is provided by a centralized chilled water plant. A dedicated air handling unit (AHU) provides the desired ventilation and air conditioning to the 5 thermal zones. It consists of a variable speed drive (VSD) supply air fan, i.e. one whose fan speed adjusts dynamically based on the required air flow and cooling demand at any given time, a cooling coil that acts as a heat exchanger, a return-air CO_2 sensor, and an outside air damper (OAD). The CO_2 measurements are used to modulate the OAD associated with the AHU to ensure that adequate fresh air is brought inside the building, which is then mixed with the return-air before being circulated to the zones. Moreover, it also helps to maintain the average indoor CO_2 concentration under the recommended 1000 ppm limit for the vast majority of time.

The building introduced above is represented in the environment as a simulation model which defines the relationship between the actions computed by the evaluated optimization algorithm, i.e. the schedule of temperature setpoints, and the variables needed for the computation of the performance measure. The latter are the zone temperature and energy costs, as defined in Subsection 5.1.5. To this end, our modelling integrates the first law of thermodynamics with regression analysis and can be represented as:

$$C_z \frac{dT_z}{dt} = \underbrace{k_a\,(T_a(t) - T_z(t))}_{\text{Heat gain from outside}} + \underbrace{k_{o,1}\theta_{CO2}(t) + k_{o,2}}_{\text{Internal heat gain from occupants}} + \underbrace{\dot{Q}_{cooling}(t)}_{\text{Zone cooling rate}} \quad (3)$$

where T_z is the zone temperature, C_z is the zone thermal capacitance, θ_{CO2} is the concentration of CO_2 in ppm denoting zonal occupancy, $k_{o,1}$ and $k_{o,2}$ are the linear regression coefficients that map θ_{CO2} to the internal heat gain due to occupancy, T_a is the outside air temperature, and k_a is the effective heat transfer coefficient between the outside and the zone. In this example, CO_2 concentration denotes the average across the 5 thermal zones since the sensor is housed in the return-air duct of the AHU. The building is not equipped with occupancy sensors at a zone level. Consequently, T_z represents the average zone temperature, i.e. of the combination of zone temperatures across the 5 thermal zones shown in Fig. 1. The zone cooling rate supplied by the HVAC is given by:

$$\dot{Q}_{cooling} = c_{p,a}[\dot{m}_s(t)(T_s(t) - T_z(t))] \tag{4}$$

which is the well-known heat transfer equation [2]. Here, $c_{p,a}$ is the specific heat capacity of air, $\dot{m}_s(t)$ is the mass flow rate of supply air to the zone, and $T_s(t)$ is the temperature of the supply air. In buildings where data for mass flow rate is unavailable, such as the one in this study, it can be approximated using the control logic of the dampers supplying the cold air. The dampers in the building rely on proportional control, modulating between closed and fully open based on the difference between the zone temperature $T_z(t)$ and the corresponding set-point temperature $T_{z,SP}(t)$. Under this approximation, we can express:

$$\dot{m}_s(t) \approx \dot{m}_{s,o} + k_c(T_z(t) - T_{z,SP}(t)) \tag{5}$$

where $\dot{m}_{s,o}$ and k_c are the coefficients corresponding to the offset and gain terms of the proportional controller.

In order to compute the required zone temperature trajectory for the time window {4 am, . . . , 5 pm} of each simulated day, (3) is used recursively to compute the zone temperature at the next time step based on the zone temperature value at current simulation time. The time step length is fixed to 5 minutes, and the initial zone temperature $T_z(4\text{ am})$ is taken from measurements at the existing building that is reflected by the environment. These measurements also include values for θ_{CO2} and T_a and are provided as part of the environment source code. The energy costs for cooling the building are approximated as:

$$E(t) = -\dot{Q}_{cooling}(t)e(t)/COP \tag{6}$$

where $e(t)$ is the electricity price at time t and COP is the coefficient of performance of the chiller, which is assumed as constant value of 2.7. The efficacy of the model is addressed in detail in [28] and omitted here for brevity. This also includes the approach for estimating the regression parameters (k_a, $k_{o,1}$, $k_{o,2}$, $\dot{m}_{s,o}$, and k_c). Without loss of generality C_z is assumed to be unity. These parameters are provided in the environment.

Finally, it should be noted that the scenario includes two sources of uncertainty that are considered relevant for the optimization, these are the development of the occupancy estimate and the outside air temperature during the day. Regarding the latter, one would typically provide a temperature forecast to the optimization algorithm, e.g. obtained from the bureau of meteorology. Naturally this forecast carries some stochastic error. However, within the scope of this work it was not possible to quantify the nature of this stochastic error; we thus provide a perfect temperature forecast, i.e. one with no error, to the evaluated algorithm. On the other hand, occupancy of the building is considered to be building specific, and it is thus left to the evaluated algorithm to account for it.

5.1.2 Relevance. Office buildings are ubiquitous, making them ideal candidates for evaluating building energy optimization algorithms. Buildings in tropical regions have relatively simple HVAC installations since weather conditions do not necessitate any heating requirements. The air conditioning system represented by this environment can thus be considered fairly representative for such buildings. Furthermore the cooling demand appears fairly representative for office buildings too, as it follows the routine of occupants arriving at work in the morning and departing from the workplace in the evening. Finally, it should be noted that the time-of-use tariff structure presented in this environment is also fairly common for commercial electricity contracts around the world.

5.1.3 Scope. The environment developed in the earlier section is amenable for solving using a variety of different optimization techniques. The work in [28] develops one such solution by first formulating a non-linear optimization problem and solving it directly using a special-purpose solver. In general, the environment gives rise to solutions that can be obtained via pure data driven approaches or by hybrid modelling techniques, i.e. one in which thermodynamics can be combined with regression analysis, as has been demonstrated in [28]. Alternatively, dynamic programming algorithms can also be applied.

5.1.4 Realism. We believe that the optimization scenario represented by our environment is realistic because it relies on sensor data readily captured by a building management system and weather forecast data that can be obtained routinely from meteorology services. Specifically, we do not rely on explicit occupancy counts being available in the thermal zones, although this information will indeed be very useful for optimizing cooling. Instead, we use CO_2 data as a proxy for occupancy. It is easy to retrofit this sensor in a building owing to the need to ensure adequate indoor air quality requirements. Moreover, CO_2 sensors are cheap and the data is inherently privacy-preserving while the use of occupancy sensors could risk leaking sensitive private information. When using the approach in a day-ahead manner, any runtime of an algorithm up to 2 hours per simulated day can be considered fast enough. Finally, the inherent assumption made in the modelling is that the chiller coefficient of performance (COP) is a constant of 2.7. This is in line with how the chillers are configured and the lack of all measurement points needed to quantify COP accurately. It is known that COP varies with time and so the assumption needs to be relaxed when appropriate chiller data needed to estimate COP accurately is known, in which case the modelling needs to be modified suitably.

5.1.5 Performance Measure. It is common for scenarios similar to ours, e.g. in [7, 33], to utilize a performance measure that balances energy costs with thermal comfort. Following this approach we define the performance measure as:

$$1 - 0.5\frac{\sum_t^T E_{ca}(t)}{\sum_t^T E_{bl}(t)} - 0.5\frac{\sum_t^T |PMV_{ca}(t)|}{\sum_t^T |PMV_{bl}(t)|} \tag{7}$$

where the summation of the terms is applied over all time steps T for all simulated days. The performance measure evenly considers

the two terms representing the improvement of energy costs $E(t)$ and thermal comfort $PMV(t)$, of the candidate algorithm *ca* relative to the baseline approach *bl*. The latter represents the rule based controller currently installed in the building which commences cooling at 7 am and proceeds to bring the zone temperatures down to 23.5 °C set-point as quickly as possible. Regarding the measure for thermal comfort, we follow common convention and use predicted mean vote (PMV) as introduced in [10], for which the best value is 0 and the comfort range is usually considered withing the range $-0.5, \ldots, 0.5$. However, PMV is computed based on various parameters depending on temperature, humidity, metabolism, etc., while we are interested in reflecting the comfort ranges defined by the facility manager as shown in (1) and (2). We thus use PMV scaled to match these comfort ranges.

Finally, it is worth noting that the performance measure reflects the improvement over the baseline strategy in percent, whereby an algorithm as good as the baseline strategy would be scored 0 and a perfect solution, i.e. one that always achieved $PMV = 0$ without producing energy costs, receives a score of 1.

5.1.6 Reproducibility. The intended usage of the environment follows an optimization of a building for which historic data about the building operation under the baseline strategy has been recorded during 57 days in spring and early summer. The task is thus to design a suitable optimization algorithm based on the provided data, which is then applied to the building and evaluated on the remaining 63 days with cooling demand in late summer and autumn. The user of the environment should thus stick to the following procedure to generate reproducible results:

```
from tropicalprecooling import TropicalPrecooling
env = TropicalPrecooling()
tested_algorithm.fit(env.get_training_data())
obs = env.reset()
done = False
while not done:
    actions = tested_algorithm.get_actions(obs)
    obs, reward, done, info = env.step(actions)
score = env.compute_performance_measure()
```

The tested_algorithm object including its functions fit and get_actions must be provided by the user.

5.1.7 Interface. Applying the current best practice, the interface follows the conventions of OpenAI's Gym [19] closely. The interaction with the environment is thus realized as calling the env.reset() and env.step() functions as illustrated in the code listing above. A detailed documentation about the objects that are provided as observations and rewards and are expected respectively as actions is omitted here for brevity. It can be found in the source code of the environment, especially in the documentation of the env.step() method.

5.2 General Applicability

It is worth noting that it was possible to identify several of the relevant characteristics for "good" environments for the evaluation of building energy optimization algorithms as our analysis of the state of the art was rather broad, i.e. also covered work related to reinforcement learning and benchmarking optimization algorithms. In fact, none of of the publications that we considered covered all relevant topics introduced in Section 4. Vice versa, this also means that all of these publications could benefit from a reevaluation of the procedure used to evaluate the performance of the respective optimization algorithms based on the findings of this work.

For example, it would certainly be beneficial for the interpretation of scores generated with the "CityLearn v1.0" environment if the documentation of it would be extended with a discussion about *relevance* and *realism* of the environment, as neither [26] nor [27] provide any details why the represented *scenario* is indeed relevant for the comparison of demand response algorithms.

Adding a discussion on *relevance* and *realism* of the utilized evaluation environments would also improve the publications reviewed in Subsection 3.2, as these topics are widely ignored in these papers. Exception to these findings are [8, 18, 20] that at least partly discourse on the *relevance* of utilized evaluation environments.

Finally, it should be noted that the examples above do not indicate that the other topics of the guide are covered in the reviewed publications to satisfactory extent. Instead the picture is more fine grained for the remaining topics, that is, the topics are partly covered in some or many of the reviewed publications. A striking example for this finding is the *performance measure*, which is explicitly defined in several papers, while it is rarely seen that a lower bound, e.g. an optimal solution, is provided to support interpretability of the reported scores.

6 CONCLUSION

This paper addresses the performance evaluation of optimization algorithms, especially in the context of building energy management. Such evaluations are routinely part of scientific publications that introduce novel or compare existing algorithms. The reported scores are commonly highly related to the utilized environments. In the context of this work, the latter are usually software components that simulate buildings or parts of those. Considering the currently common practices of how environments are developed and used, the reported scores appear of questionable significance, as e.g. environments are not developed independently of optimization algorithms or reflect only one arbitrarily chosen building. To overcome this issue it has been suggested to establish a collection of shared environments as a common foundation for the objective evaluation of building energy optimization algorithms. However, such a collection will certainly only be of use if the individual members are well designed w.r.t. the goal of evaluating building energy optimization algorithms, which is why the design of suitable environments is addressed in this work.

The foundation of our work is a detailed analysis of related publications, as documented in Section 3, that allowed the identification of topics that are considered relevant for environments. These are: *scenario*, *relevance*, *scope*, *realism*, *performance measure*, *reproducibility* and *interface*. Going on, a practical guide has been developed that is the subject of Section 4, providing guiding questions and explanations of these for each topic, and also reasoning why the respective points are considered relevant. A brief overview of the essential points is given in Table 1. Finally, Section 5 covers the practical application of the guide, by introducing a novel environment but also discussing potential benefits for existing publications

if the guide would be applied to improve the proposed or utilized evaluation scenarios. Through this it was possible to demonstrate that the guide is indeed a useful tool for the design of environments for the evaluation of building energy optimization algorithms, but also to provide a complete example for a scientific documentation of an environment. The result of this design process, the Tropical-Precooling environment, reflects the optimization of energy costs for cooling an office building in tropical climate. It is published alongside this publication and a link to access the source code is provided in the section below.

To conclude the above we consider our work as a valuable stepping stone towards a collection of shared environments for the evaluation of building energy optimization algorithms, thus supporting the development of better and more general algorithms for improving the sustainability of buildings.

SUPPLEMENTARY MATERIAL

The source code of the TropicalPrecooling environment including further documentation can be found at: https://github.com/fzi-forschungszentrum-informatik/tropical_precooling_environment

ACKNOWLEDGMENTS

The authors would like to thank the Townsville City Council for providing the data and information about the building that has been used as blueprint for the TropicalPrecooling environmet. This research has partly been funded by the German Federal Ministry for Economic Affairs and Energy within the projekt "Flexkälte – Flexibilisierung vorhandener Kälteanlagen und deren optimierter Einsatz in einer Realweltanwendung".

REFERENCES

[1] Álvarez, J. D., Redondo, J. L., Camponogara, E., Normey-Rico, J., Berenguel, M., and Ortigosa, P. M. Optimizing building comfort temperature regulation via model predictive control. *Energy and Buildings* (2013).

[2] Balaji, B., Teraoka, H., Gupta, R., and Agarwal, Y. Zonepac: Zonal power estimation and control via HVAC metering and occupant feedback. In *Proc. ACM BuildSys* (Italy, 2013).

[3] Beiranvand, V., Hare, W., and Lucet, Y. Best practices for comparing optimization algorithms. *Optimization and Engineering 18*, 4 (2017), 815–848.

[4] Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research 47* (jun 2013), 253–279.

[5] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. OpenAI Gym, 2016.

[6] Chen, B., Cai, Z., and Bergés, M. Gnu-RL: A precocial reinforcement learning solution for building HVAC control using a differentiable MPC policy. *BuildSys 2019 - Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2019), 316–325.

[7] Ding, X., Du, W., and Cerpa, A. OCTOPUS: Deep reinforcement learning for holistic smart building control. *BuildSys 2019 - Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2019), 326–335.

[8] Doukas, H., Patlitzianas, K. D., Iatropoulos, K., and Psarras, J. Intelligent building energy management system using rule sets. *Building and Environment* (2007).

[9] Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. Benchmarking Deep Reinforcement Learning for Continuous Control. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48* (2016), ICML'16, JMLR.org, pp. 1329–1338.

[10] Fanger, P. O. *Thermal comfort. Analysis and applications in environmental engineering.* Copenhagen: Danish Technical Press., 1970.

[11] Hansen, N., Finck, S., Ros, R., and Auger, A. Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions. Research Report RR-6829, INRIA, 2009.

[12] Kastner, W., Kofler, M. J., and Reinisch, C. Using AI to realize energy efficient yet comfortable smart homes. *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS* (2010), 169–172.

[13] Machado, M. C., Bellemare, M. G., Talvitie, E., Veness, J., Hausknecht, M., and Bowling, M. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research 61* (2018), 523–562.

[14] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning.

[15] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature 518*, 7540 (feb 2015), 529–533.

[16] Molina, D., Lu, C., Sherman, V., and Harley, R. G. Model predictive and genetic algorithm-based optimization of residential temperature control in the presence of time-varying electricity prices. *IEEE Transactions on Industry Applications 49*, 3 (2013), 1137–1145.

[17] Moriyama, T., De Magistris, G., Tatsubori, M., Pham, T.-H., Munawar, A., and Tachibana, R. Reinforcement Learning Testbed for Power-Consumption Optimization. In *Methods and Applications for Modeling and Simulation of Complex Systems* (Singapore, 2018), Springer Singapore, pp. 45–59.

[18] Oldewurtel, F., Parisio, A., Jones, C. N., Gyalistras, D., Gwerder, M., Stauch, V., Lehmann, B., and Morari, M. Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy and Buildings 45* (2012), 15–27.

[19] OpenAI. openai/gym: A toolkit for developing and comparing reinforcement learning algorithms., 2020.

[20] Salpakari, J., and Lund, P. Optimal and rule-based control strategies for energy flexibility in buildings with PV. *Applied Energy 161* (2016).

[21] Schibuola, L., Scarpa, M., and Tambani, C. Demand response management by means of heat pumps controlled via real time pricing. *Energy and Buildings 90* (2015).

[22] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal Policy Optimization Algorithms.

[23] Shaikh, P. H., Nor, N. B. M., Nallagownden, P., Elamvazuthi, I., and Ibrahim, T. A review on optimized control systems for building energy and comfort management of smart sustainable buildings. *Renewable and Sustainable Energy Reviews 34* (jun 2014), 409–429.

[24] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of Go with deep neural networks and tree search. *Nature 529*, 7587 (jan 2016), 484–489.

[25] Sutton, R. S., and Barto, A. *Reinforcement Learning - An Introduction*, second ed. 2018.

[26] Vázquez-Canteli, J. R., Kämpf, J., Henze, G., and Nagy, Z. CityLearn v1.0: An OpenAI gym environment for demand response with deep reinforcement learning. *BuildSys 2019 - Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2019), 356–357.

[27] Vázquez-Canteli, J. R., Nagy, Z., Dey, S., and Henze, G. The CityLearn Challange - Project website, 2020.

[28] Vishwanath, A., Chandan, V., Mendoza, C., and Blake, C. A data driven pre-cooling framework for energy cost optimization in commercial buildings. *Proc. ACM e-Energy* (2017), 157–167.

[29] Waibel, C., Mavromatidis, G., Evins, R., and Carmeliet, J. A comparison of building energy optimization problems and mathematical test functions using static fitness landscape analysis. *Journal of Building Performance Simulation 12*, 6 (2019), 789–811.

[30] Wang, Z., and Wang, L. Intelligent control of ventilation system for energy-efficient buildings with CO2 predictive model. *IEEE Transactions on Smart Grid 4*, 2 (2013), 686–693.

[31] Wölfle, D., Förderer, K., and Schmeck, H. A Concept for Standardized Benchmarks for the Evaluation of Control Strategies for Building Energy Management. *Energy Informatics 2*, Suppl 2:P3 (2019).

[32] Zhang, C., Kuppannagari, S. R., Kannan, R., and Prasanna, V. K. Building HVAC scheduling using reinforcement learning via neural network based model approximation. *BuildSys 2019 - Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2019), 287–296.

[33] Zhang, Z., and Lam, K. P. Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system. *BuildSys 2018 - Proceedings of the 5th Conference on Systems for Built Environments* (2018), 148–157.