Greener Bits: Formal Analysis of Demand Response

Christel Baier[1], Sascha Klüppelholz[1], Hermann de Meer[2], Florian Niedermeier[2], and Sascha Wunderlich[1] *

[1] Technische Universität Dresden
{baier,klueppel,wunder}@tcs.inf.tu-dresden.de
[2] University of Passau
{hermann.demeer,florian.niedermeier}@uni-passau.de

Abstract. Demand response is a promising approach to deal with the emerging power generation fluctuations introduced by the increasing amount of renewable energy sources fed into the grid. Consumers need to be able to adapt their energy consumption with respect to the given demand pattern and at the same time ensure that their adaptation (i.e., response) does not interfere with their various operational objectives. Finding, evaluating and verifying adaptation strategies which aim to be optimal w.r.t. multiple criteria is a challenging task and is currently mainly addressed by hand, heuristics or guided simulation. In this paper we carry out a case study of a demand response system with an energy adaptive data center on the consumer side for which we propose a formal model and perform a quantitative system analysis using probabilistic model checking. Our first contribution is a fine-grained formal model and the identification of significant properties and quantitative measures (e.g., expected energy consumption, average workload or total penalties for violating adaptation contracts) that are relevant for the data center as an adaptive consumer. The formal model can serve as a starting point for the application of different formal analysis methods. The second contribution is an evaluation of our approach using the prominent model checker PRISM. We report on the experimental results computing various functional properties and quantitative measures that yield important insights into the viability of given adaptation strategies and how to find close-to-optimal strategies.

1 Introduction

In modern society, a permanent and reliable availability of electrical power has become an indispensable good for many aspects of life. However, the continuously

* The authors are supported by the DFG through the Collaborative Research Center SFB 912 – HAEC, the Excellence Initiative by the German Federal and State Governments (cluster of excellence cfAED and Institutional Strategy), the Research Training Groups QuantLA (GRK 1763) and RoSI (GRK 1907), and by the EU through the European Union's Seventh Framework Programme FP7 in the context of the DC4Cities project (grant agreement no. 609304).

increasing demand for electrical energy comes at a price: guaranteeing stability under almost any load condition requires a constant adaptation of power generation to keep production and volatile demand in equilibrium. However, power plants capable of quickly adapting their power output to balance the grid are usually driven by fossil fuels and therefore carbon-intensive. From an environmental perspective, this is highly undesirable and an increased feed-in of renewable energy is encouraged. Energy sources like wind or solar are exhibiting volatile availability patterns and can therefore not provide the same balancing capabilities as fossil fuel based power plants. A promising approach to cope with these shortcomings is demand response (DR) exerting control over power demand on the consumer side. DR is widely recognized as a promising approach to reduce the costs for mitigating operational instability of power grids when incorporating renewable energy sources, but standardization is still underway [5]. Many recent works focus on the design of DR-programs that define the communication protocol between power producers and power consumers. Current DR-programs can be classified into three categories: price-based, incentive-/event-based and demand reduction bids (see, e.g., [29]).

For participants in a DR-system to be a valuable asset to a distribution system operator, the adaptation to a DR-request needs to be enacted reliably. At the same time, the respective participant needs to ensure the power adaptation does not interfere with operational objectives. The quantitative impact of a DR-interaction on power demand and operational performance on the consumer side needs to be foreseeable and is well suited for analysis via formal methods, as it may impact critical processes on both power grid and consumer side. Data centers are particularly well-suited for participation in DR as they consume large amounts of energy and provide many opportunities in dynamically adapting to external demands by applying advanced resource and workload management strategies. Data center DR-systems have been considered from several perspectives, e.g., pricing [22, 31], implementation [10], communication [6], and contract design [7]. Working prototypes of DR-systems for data centers were implemented in EU FP7 projects ALL4Green [6] and DC4Cities [20, 25], where the latter focuses on the possibility of continuously adapting to a given power plan.

Despite wide recognition of the potential of DR, data centers are currently hardly participating in DR, mainly due to the fact that the design of efficient adaptation strategies is a non-trivial task. The goal of this paper is to show that formal methods can contribute to this task, e. g. , by providing guarantees on cost/utility requirements in worst-case scenarios and by evaluating existing resource management strategies to gain insights for the design of efficient scheduling strategies.

Contribution. In this paper we provide a detailed formal model for DR-systems with data centers on the consumer side. The model is compositional and uses Markov decision processes (MDPs) equipped with reward functions to capture various quantitative measures. The choices in the MDP stand for the possible workload scheduling. The base model consists of components for the data center, a request generator, the service load, and a component for green energy forecast.

For the sake of simplicity we assume a simple incentive-based DR-program and present possible extensions that allow for addressing alternative and more complex DR-scenarios by refining the components of the base model. For the analysis we use probabilistic model checking (PMC) to compute minimal and maximal probabilities and expectations that provide guarantees in worst-case scenarios as well as insights for the design of efficient scheduling strategies maximizing or minimizing various performance and costs objectives. The class of considered measures subsumes conventional cost-/utility indices important for DR-systems (e.g., *Power Usage Efficiency* (PUE) [28], and *Energy-Response Time Product* (ERP) [15]). We illustrate the feasibility of the modeling and analysis approach in experimental studies on the base model and report here on the scalability and insights gained in the process. The experiments were carried out using extensions of the prominent probabilistic model checker PRISM [21].

Related work. We are not aware of any work describing a quantitative analysis of DR-systems using PMC. Currently, load adjustment in data centers under DR-programs is usually not formally modeled or analyzed. Planning and verification in power grids is often performed via special simulators (e.g., PowerWorld Simulator [18]). However, these are mostly used for strategic decisions on power grid development in the long term or to find solutions to the question which electrical loads to shed in an emergency (see, e.g., [9,16,23]). The latter problem is closely related to demand response, however loads are assigned priorities and decisions on power generation side are made only in case of emergencies. Other existing work focuses on coordination between different energy sources (e.g., [30]) with the goal to optimize a performance index such as PUE or ERP. Solutions are found by solving equality constrained optimization problems or mixed integer linear programs and then evaluated using simulation of large real-world workload traces and current energy prices (e.g., [11,30,31]). Other formal approaches in the context of DR address, e. g. , the stability of a given Markov model [8] or provide uncertainty models in which Markov chains are combined with additional random transition matrices [24]. Model checking and in particular PMC has been applied to related problems, e.g., for energy-aware task scheduling [17] and dynamic power management [26], but also for controller synthesis (e.g., [13]).

2 Scenario

Especially in future smart grid scenarios, in which high amounts of renewable power generation are to be expected, both events of energy surplus and scarcity have to be considered as the controllability of many renewable sources is very limited. Therefore, demand response requests (DR-requests) are considered as a mechanism to trigger increasing or decreasing power demands. In this work, we visit a demand response scenario, in which an energy management authority is creating DR-requests for power adaptation in order to influence the power demand on consumer side. In the following, we will assume a demand response request to contain (1) the start and the end of the adaptation time-interval and (2) a target power demand range during this time period.

A DR-request may arrive at the consumer side at any point in time (typically probabilities are non-uniformly distributed). Failing to adhere to the power bounds during the demand response interval will cause a penalty. Figure 1 shows the high level interaction diagram in the assumed DR-system.

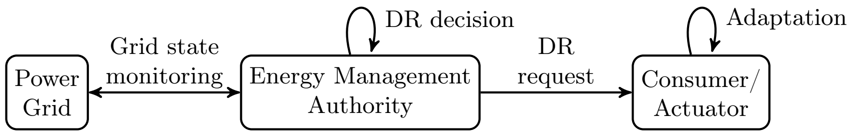


Fig. 1: High level DR-system overview.

Consumer side. On the consumer side, we assume that parts of the load can be shifted for a certain time. For simplicity we assume that processing workload causes a proportional increase in the power demand. In general, one needs to be aware of the correlation between workload and the corresponding power demand, which is in turn directly influenced by the (re)scheduling of workload. To reliably reach the power demand requested in the DR-request workload has to be rescheduled in such a way as to reach target power demand, while at the same time ensuring, e.g., that all work is scheduled until a given deadline. Here, we assume that all workload has to be processed until the end of a day. Failing to schedule workload until the deadline will result in a (load-specific) penalty. Apart from work which can be rescheduled (batch load), we assume that certain work (service load) has to be immediately processed. The fraction of workload to be processed immediately is assumed to be varying in time, however always less than the total load capacity. This type of workload can neither be rescheduled nor canceled. Figure 2 shows an example of the rescheduling process.

Additionally, we assume that - in line with our future smart grid setting - the consumer side is equipped with local renewable energy generation capabilities (e.g., small wind turbine or solar panels). The power generated at different points in time can be forecasted by utilizing weather data and/or information on historical generation. However, due to possible fluctuation (e.g., solar radiation on cloudy days), forecasting errors are common and will cause deviations of actual generation from the forecast values.

Objectives. Generally, objectives may fall into one of the following categories. One can either optimize cost or utility measures (1) on the consumer side, (2) on the energy management authority side, or (3) on both sides given certain

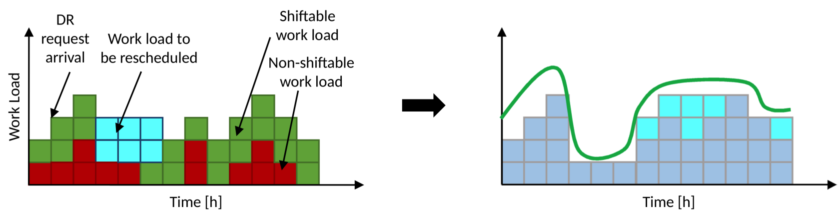


Fig. 2: Example adaptation of a flexible load.

additional constraints. In this paper we focus on the consumer side, but the methods are also applicable for optimization on the energy management authority side and mixtures thereof.

Data centers. A specific use case of demand response are data centers. Participation of data centers within demand response systems is highly attractive, as they have automation frameworks already in place, making automated processing of demand response requests possible. Additionally, data centers consume large amounts of electrical energy and therefore are well suited to create a significant impact when adapting their power demand. We assume data centers to process two basic types of load: interactive and batch. Interactive load is characterized by service level agreements which require strict bounds on response time for users to have a high quality of experience. Therefore interactive load cannot be rescheduled and has to be processed immediately. Typical examples include web servers, stock traders and virtualized desktop environments. In contrast, batch load may be processed at any time while completing before its deadline. Batch jobs may therefore be arranged in a way as to adapt data center power demand according to demand response requests.

3 Theoretical foundations

Throughout the paper, we assume the reader is familiar with Markovian models. A brief summary of the relevant concepts for Markov decision processes is provided below. For more details, we refer to, e. g. , [27].

Markov decision processes. An MDP is a tuple $\mathcal{M} = (S, \mathrm{Act}, P, \mathrm{AP}, \mathsf{L})$, with a finite set of states S, a finite set of actions Act, transition probabilities $P : S \times \mathrm{Act} \times S \to [0, 1]$, a finite set of atomic propositions AP and a labeling function $\mathsf{L} : S \to 2^{\mathrm{AP}}$. We require that the values $P(s, \alpha, s')$ are rational and $\sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\}$ for all states $s \in S$ and actions $\alpha \in \mathrm{Act}$. The triples (s, α, s') with $P(s, \alpha, s') > 0$ are called *transitions*. Action α is said to be *enabled* in state s if $P(s, \alpha, s') > 0$ for some state s'. $\mathrm{Act}(s)$ denotes the set of actions that are enabled in $s \in S$. To avoid terminal behaviors, we require that $\mathrm{Act}(s) \neq \varnothing$ for all states s. Paths in an MDP $\mathcal{M}$ can be seen as sample runs. Formally, they are finite or infinite sequences where states and actions alternate, i. e. , $\pi = s_0\,\alpha_0\,s_1\,\alpha_1 \ldots \in (S \times \mathrm{Act})^* S \cup (S \times \mathrm{Act})^\omega$ with $\alpha_i \in \mathrm{Act}(s_i)$ and $P(s_i, \alpha_i, s_{i+1}) > 0$ for all i. In the following, we assume that an initial state s is given. For a path property ϕ, we write $\mathrm{Pr}^{\sigma}_{\mathcal{M}}(\phi)$ for the probability of ϕ in $\mathcal{M}$ under scheduler σ. Additionally, we write $\mathrm{Pr}^{\min}_{\mathcal{M}}(\phi)$ and $\mathrm{Pr}^{\max}_{\mathcal{M}}(\phi)$ for the minimal and maximal probabilities for ϕ among all schedulers σ. In case the action set Act is a singleton it can be omitted, since the behavior is then completely deterministic and the MDP degenerates to a Markov chain (MC).

Reward functions. A reward function $\mathrm{rew} : S \times \mathrm{Act} \to \mathbb{N}$, annotates state-action pairs with a natural number. Each reward function can be lifted to assign to each finite path its accumulated value $\mathrm{rew}(s_0\alpha_0 s_1\alpha_1 \ldots s_n\alpha_n) = \sum_{i=0}^{n-1} \mathrm{rew}(s_i, \alpha_i)$. For a set of states G and a scheduler σ such that $\mathrm{Pr}^{\sigma}_{\mathcal{M}}(\Diamond\, G) = 1$ we can then introduce the expected reward until reaching G. To define the expected reward, we let P^r_G be the set of paths $\pi = s_0\alpha_0 s_1 \ldots$ such that there exists an $n \in \mathbb{N}$ with $s_n \in G$, $s_i \notin G$ for all $i < n$ and $\mathrm{rew}(s_0 \ldots s_n) = r$. Then $\mathrm{Ex}^\sigma_\mathcal{M}[\mathrm{rew}](\lozenge\, G) = \sum_{r=0}^{\infty} r \cdot \mathrm{Pr}^\sigma_\mathcal{M}(P^r_G)$. As before we define the extremal expectations $\mathrm{Ex}^{\min}_\mathcal{M}$ and $\mathrm{Ex}^{\max}_\mathcal{M}$ for minimizing and maximizing schedulers.

4 Formal model

In this section we present an MDP-based compositional model for demand response formalized for the use case of a data center. As described in Section 2 the scenario consists of the data center on the consumer side and an energy authority sending DR-requests. The data center needs to schedule batch work and interactive work, it reacts to DR-requests and can have additional green energy sources available. In our model those influences are formalized using stochastic distributions and the nondeterministic choices within the data center constitute an MDP model that yields the basis for further formal analysis. Based on this setting, the goal is to find appropriate adaptation strategies (i.e., resolving the nondeterminism in the MDP) that optimize for various cost/utility objectives, as introduced later in this section.

In our model we fix the number of batch work jobs $J_0 \in \mathbb{N}$ that the data center can schedule over one day, as well as a number $T \in \mathbb{N}$ of discrete time steps into which the day is divided, which can be seen as the resolution of the time domain. At every time point the data center can decide which batch work should be scheduled next. The maximal number of simultaneous jobs is fixed as $\mathsf{capacity} \in \mathbb{N}$. The data center's scheduling decision is influenced by the amount of interactive work (service load), the available energy and the received DR-requests from the energy authority, which can arrive at any time. In the base model it is assumed that requests can not be refused. Furthermore, all jobs and services are already in the shape of an independent *least schedulable unit (LSU)*, i.e., work packages which cannot be interrupted, have no dependencies and require one energy unit over its lifetime.

The model is equipped with simple reward functions to capture the amount of green energy that was produced, the amount of brown energy that had to be bought, the number of time steps, and penalties for violating DR-requests. For the latter, rather simple functions are used, but in general one could use arbitrary complex functions to capture penalties.

4.1 Component model

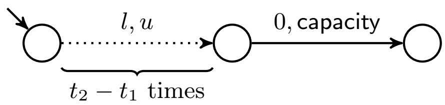
The base model consists of the following four components, each represented as individual Markov chains or MDPs: stochastic service load and DR-request generators $\mathcal{M}_{\mathsf{serv}}$ and $\mathcal{M}_{\mathsf{req}}$, a green-energy forecast $\mathcal{M}_{\mathsf{fc}}$ and the data center itself $\mathcal{M}_{\mathsf{dc}}$. From those components one large MDP is then composed for the composite model, i.e., $\mathcal{M} = \mathcal{M}_{\mathsf{serv}} \otimes \mathcal{M}_{\mathsf{fc}} \otimes \mathcal{M}_{\mathsf{req}} \otimes \mathcal{M}_{\mathsf{dc}}$.

The compositional modeling approach allows to easily generate variants, e. g. , with less, more or other participants and hence it facilitates the maintainability of the model. In Section 4.2 we will discuss some possible refinements and extensions of our base model. We now consider the details of the four base components.

$\mathcal{M}_{\mathsf{serv}}$ **- service load generator.** The service load of a data center is assumed to be stochastically distributed. For any point in time $t \in \{ 0, 1, \ldots, T \}$ a random variable $\mathsf{service}(t) \in \{ 0, 1, \ldots, \mathsf{capacity} \}$ is given, signifying the number of interactive jobs to be executed at time t. From this, a Markov chain $\mathcal{M}_{\mathsf{serv}}$ can be derived, with states of the form (t, c) for $c = \mathsf{service}(t)$. The probability of a transition $(t, c) \to (t + 1, c')$ in $\mathcal{M}_{\mathsf{serv}}$ is the probability of $\mathsf{service}(t + 1)(c')$.

$\mathcal{M}_{\mathsf{fc}}$ **- green-energy forecast.** To reflect the probabilistic green-energy production, another random variable $\mathsf{produce}(t) \in \{ 0, 1, \ldots, \mathsf{prodmax} \}$ is introduced for each point in time $t \in \{ 0, 1, \ldots, T \}$. Intuitively, this variable represents the possible energy production values at time t, which may depend on the weather, season or time of day. Similar to the service load, a Markov chain $\mathcal{M}_{\mathsf{fc}}$ with states (t, e) for $e = \mathsf{produce}(t)$ can be derived. The probability for $(t, e) \to (t + 1, e')$ is the probability of $\mathsf{produce}(t + 1)(e')$.

$\mathcal{M}_{\mathsf{req}}$ **- request generator.** We are given statistical information on the arrival of DR-requests and their format. A DR-request arriving at some time point t with rate $r \in [0, 1]$ is represented by a triple (I, l, u) with a discrete time interval $I \subseteq [t, T]$ and lower and upper resource bounds $l, u \in \mathbb{N}$ such that $0 \le l \le u \le \mathsf{capacity}$. Intuitively, a DR-request (I, l, u) signifies that at each time point $t \in I$ the resource requirements should be between l and u. Hence, the number of jobs executed at time t should be in that interval. A single DR-request $R_i = ([t_1, t_2], l, u)$ can be modeled as a degenerated Markov chain (i. e., all probabilities are 1) $\mathcal{A}_i$ as follows:



The request generator then has an initial state with outgoing transitions to each DR-request R_i with probability r_i and a self-loop with probability $1 - \sum_{i=0}^{n} r_i$.

$\mathcal{M}_{\mathsf{dc}}$ **- data center.** The data center keeps track of the current time step value $t \in \{ 0, 1, \ldots, T \}$ and the number J of jobs that are still to be processed. Initially at time point $t = 0$, $J = J_0$ and J will decrease until either $J = 0$ or the day is over, i.e., $t = T$. At each point in time, the data center can choose to schedule a number $j \le \min\{ J, \mathsf{capacity} \}$ of jobs. These choices are modeled nondeterministically. Whether or not the action of choosing j jobs at time point t will be enabled depends on the produced service load $\mathsf{service}(t)$. Hence, the enabled actions in the composite model $\mathcal{M}$ will be the following. Let $s = (t, c, e, l, u, J)$ be a state of $\mathcal{M}$. Then, the action of choosing $j \in \mathbb{N}$ jobs should be enabled in $\mathcal{M}$ iff $c + j \le \min\{ J, \mathsf{capacity} \}$, i.e., iff the interactive load plus the number of scheduled jobs is not larger than the capacity and enough jobs are still available.

Penalties and other reward functions. For modeling the costs of violating a DR-request at time t we introduce a reward function $\mathsf{penalty}$. The penalty to be paid in state $s = (t, c, e, l, u, J)$ of $\mathcal{M}$ when scheduling $j \in \mathbb{N}$ jobs is defined as

$$\mathsf{penalty}(s, j) = \mathsf{lpenalty}(s, j) + \mathsf{upenalty}(s, j)$$

i.e., the sum of a penalty for violating the lower or upper bound given as two separate reward functions defined as follows:

$$\mathsf{lpenalty}(s, j) = \max\{\ c + j - l, 0\ \} \qquad \mathsf{upenalty}(s, j) = \max\{\ u - c - j, 0\ \}$$

for state is $s = (t, c, e, l, u, J)$ in $\mathcal{M}$. In general, the penalty function (and other cost/utility functions) can be nearly any complex arithmetic expression over variables in the model. In particular, the reward functions do not affect the state space of $\mathcal{M}$ and hence do not contribute to the complexity of the model.

Besides the reward function for penalties we introduce reward functions that can then be used inside formulas for various objectives as detailed in Section 4.3. Specifically, we use $\#\mathsf{steps}$, $\#\mathsf{jobs}$ and $\#\mathsf{requests}$ for the number of time steps, finished jobs and accepted DR requests. The latter is only relevant in the model variants where the data center can refuse incoming DR-requests. Furthermore, green signifies the produced green energy and brown the bought grid energy.

4.2 Model variants

In the following we introduce several variants which allow modifying the scenario to be considered by replacing components of the compositional model. More variants can be found in the extended version [4].

Heterogenous jobs and dependencies. Instead of assuming that every job can be decomposed into uniform LSUs, we can introduce a more general case. There, each job $j \in J \subseteq \mathbb{N} \times \mathbb{N}$ carries a length and a energy-per-time-unit value. The data center component is then more complicated. Intuitively, each state in $\mathcal{M}_{\mathsf{dc}}$ now carries two pieces of information: a set W of jobs currently worked on and a set O of jobs which are still pending. The energy consumed in a state is the sum of the energy-per-time-unit values of the jobs in its working set W. Outgoing transitions of a state s in $\mathcal{M}_{\mathsf{dc}}$ are then labeled by a subset $A \subseteq O$ of open jobs and lead to a state s', in which A becomes the set of jobs currently worked on and $O \setminus A$ becomes the new set of open jobs. To allow dependencies among jobs, they are partially ordered in a set $(J, \leq)$ where $j_1 \leq j_2$ if j_1 has to be completed before j_2 can be scheduled. This variant introduces a combinatorial blowup in the data center component and hence the composed model.

Hard limits. It is possible to represent hard limits on the DR-requests, i. e., for a request (I, l, u) to disallow using less energy than l or more energy than u while the request is active. This can be modeled by modifying the enabled actions of $\mathcal{M}$ as follows. In a state $s = (t, c, l, u, J)$, the action j is enabled iff $c + j \leq \min\{\ J, \mathsf{capacity}\ \}$ as before and additionally $l \leq c + j \leq u$ must hold.

Accepting and refusing DR-requests. Instead of forcing DR-request, we may equip the data center with additional non-deterministic choices for accepting or refusing DR-requests. Adaptation strategies for the data center may then refuse a

DR-request by not scheduling the corresponding action. In this setting additional reward functions are of interest, e.g., a reward function for tracking the bounties for accepted DR-requests.

Adding an adaptation strategy. Another important variant allows for analyzing specific adaptation strategies. This way, a consumer can formally evaluate currently implemented strategies with respect to various objectives. This amounts to adding a (possibly randomized) scheduler that resolves the nondeterministic choices in $\mathcal{M}_{\mathsf{dc}}$, resulting in a Markov chain $\mathcal{M}'$. One can then compare the results for $\mathcal{M}'$ with theoretically optimal strategies in $\mathcal{M}$ w.r.t. a given objective.

Multiple data centers and different DR-protocols. It is possible to introduce copies and variants of $\mathcal{M}_{\mathsf{dc}}$ to model multiple data centers. DR-protocols which do not require interaction between the energy authority and the data center like incentive-based and price-based ones can be modeled by modifying the enabled actions or by introducing further reward functions.

4.3 Objectives

In this section we introduce different kinds of evaluation criteria that are important in the given setting, in particular for optimizing various cost and performance measures. We illustrate their relevance for demand response with example objectives formulated for the data center scenario. For the formulas, we use the usual temporal operators $\Diamond$ (eventually) and $\Box$ (always).

The first class of objectives is concerned with the confidence in our model. Model checking of such purely functional properties can be applied in addition to, e.g., simulation of the model. Typically, one is concerned with whether the minimal or maximal probability of certain temporal events is either zero or one, or with probabilities and expected values for costs/utility being within reasonable bounds. We present here a few examples that can be computed using standard PMC-methods. E.g., to search for unintended deadlocks, one can check whether the minimal probability of reaching the end of the day (eod) is one, as then no scheduler can avoid reaching the end of the day.

$$\mathrm{Pr}^{\min}_{\mathcal{M}}(\Diamond\, \mathsf{eod}) = 1 \tag{1}$$

The above formula can be enriched, e. g. , with a step bound, to ensure that the end of the day will be reached within the desired number of steps.

$$\mathrm{Pr}^{\min}_{\mathcal{M}}(\Diamond^{\#\mathsf{steps}=T}\ \mathsf{eod}) = 1 \tag{2}$$

It is also of interest to compute the probability to finish all jobs using an optimal scheduler and to check whether the result is within reasonable boundaries.

$$\mathrm{Pr}^{\max}_{\mathcal{M}}(\Diamond\,(J = 0)) \tag{3}$$

Furthermore, the maximal probability for surviving the day without using brown energy (and to complete all jobs) is significant, although in general very low.

$$\mathrm{Pr}^{\max}_{\mathcal{M}}(\Box\ \mathsf{green\_only}) \tag{4}$$

$$\mathrm{Pr}^{\max}_{\mathcal{M}}(\Box\ \mathsf{green\_only} \wedge \Diamond\,(J = 0)) \tag{5}$$

The atomic proposition $\mathsf{green\_only}$ signifies that no brown energy was used.

The second important class of objectives concerns the optimization of a single quantitative measure addressing either cost or utility, both either from the consumer perspective or the energy authority perspective. Within this class of properties we cannot address trade-offs. As utility measures one could, for example, compute the probability for finishing at least $n \in \mathbb{N}$ jobs by the end of the day, or the expected numbers of jobs that could be finished by the end of the day assuming optimal schedulers.

$$\Pr^{\max}_{\mathcal{M}}(\Diamond^{\mathsf{\#jobs} \geq n}\ \mathsf{eod}) \tag{6}$$
$$\mathrm{Ex}^{\max}_{\mathcal{M}}[\mathsf{\#jobs}](\Diamond\, \mathsf{eod}) \tag{7}$$

On the cost side there is, e. g. , the probability of surviving the day when the amount of brown energy used is bounded by $n \in \mathbb{N}$, or the minimal and maximal expected penalty when a DR-request was not fulfilled either until the end of the day or until all jobs are done.

$$\Pr^{\max}_{\mathcal{M}}(\Diamond^{\mathsf{brown} \leq n}\ \mathsf{eod}) \tag{8}$$
$$\mathrm{Ex}^{\max}_{\mathcal{M}}[\mathsf{penalty}]\,(\Diamond\ \mathsf{eod}) \tag{9}$$
$$\mathrm{Ex}^{\min}_{\mathcal{M}}[\mathsf{penalty}]\,(\Diamond\ \mathsf{eod}) \tag{10}$$
$$\mathrm{Ex}^{\max}_{\mathcal{M}}[\mathsf{penalty}]\,(\Diamond\ (J = 0)) \tag{11}$$
$$\mathrm{Ex}^{\min}_{\mathcal{M}}[\mathsf{penalty}]\,(\Diamond\ (J = 0)) \tag{12}$$

Another important cost measure is the minimal and maximal expected time (number of steps) until all jobs are done or the end of the day has been reached.

$$\mathrm{Ex}^{\min}_{\mathcal{M}}[\mathsf{\#steps}]\,(\Diamond((J = 0)\ \vee\ \mathsf{eod})) \tag{13}$$
$$\mathrm{Ex}^{\max}_{\mathcal{M}}[\mathsf{\#steps}]\,(\Diamond((J = 0)\ \vee\ \mathsf{eod})) \tag{14}$$

Considering utility and cost measures, one is typically interested in their trade-off, ideally (although impossible) for maximizing the utility and minimizing the cost at the same time. In the following we will consider quantiles and conditional probabilities as important instances of this class and illustrate their relevance in the demand response setting again with a few examples.

Quantile queries ask for the maximal or minimal value of a variable such that the probability threshold for a property is still within a defined range. They can be computed by the techniques in [1]. Interesting quantiles are, e. g. , the maximum number of jobs that can be finished within one day with probability at least $p \in (0, 1)$ or the maximum number of DR-requests that can be accepted such that the probability of finishing all jobs by the end of the day is sufficiently high. An analogous quantile can be formulated considering the minimal penalty rather than the maximum number of DR-requests.

$$\max\{\ j \in \mathbb{N} : \Pr^{\max}_{\mathcal{M}}(\Diamond^{\mathsf{\#jobs} \geq j}\ \mathsf{eod}) \geq p\ \} \tag{15}$$
$$\max\{\ r \in \mathbb{N} : \Pr^{\max}_{\mathcal{M}}(\Diamond^{\mathsf{\#requests} < r}\ (J = 0)) > p\ \} \tag{16}$$
$$\min\{\ y \in \mathbb{N} : \Pr^{\max}_{\mathcal{M}}(\Diamond^{\mathsf{penalty} \leq y}\ (J = 0)) > p\ \} \tag{17}$$

An alternative way of combining multiple simple measures are conditional probabilities in which one measure serves as condition and another serves as the objective of interest. They can be solved using the techniques in [2].

$$\Pr^{\max}_{\mathcal{M}}(\Box\, \mathsf{green\_only}\ |\ \Diamond\, (J = 0)) \tag{18}$$
$$\Pr^{\max}_{\mathcal{M}}(\Diamond\, (J = 0)\ |\ \Box\, \mathsf{green\_only}) \tag{19}$$

This formula queries the maximum probability for consuming green energy only given the condition that all jobs will be finished eventually. In Formula (19) the roles of the objective of interest and the condition are swapped. The following are formulas conditional versions of Formulas (8) and (6).

$$\Pr^{\max}_{\mathcal{M}}(\Diamond^{\mathsf{brown} \leq n}\ \mathsf{eod} \mid \Box\,(J > 0)) \qquad \Pr^{\max}_{\mathcal{M}}(\Diamond^{\#\mathsf{jobs} \geq n}\ \mathsf{eod} \mid \Box\ \mathsf{green\_only})$$

Additional objectives. While the above values can be computed with standard PMC-techniques, the following objectives require special PMC-algorithms.

A simple objective is to finish all queued jobs with minimal expected accumulated penalty. Formally, one wants to find a scheduler σ for $\mathcal{M}$ such that $\Pr^{\sigma}_{\mathcal{M}}(\Diamond\,(J = 0)) = 1$ and the expected accumulated penalty is minimal among all schedulers τ with $\Pr^{\tau}_{\mathcal{M}}(\Diamond\,(J = 0)) = 1$. This task can be solved by the techniques in [14]. Another example objective for which no PMC-methods are available so far is to find the minimal q and a scheduler σ such that $\Pr^{\sigma}_{\mathcal{M}}(\Diamond\,(J = 0)) = 1$ and the penalty is at most q in each step, i. e., $\Pr^{\sigma}_{\mathcal{M}}\,(\Box(\mathsf{penalty} \leq q) \wedge \Diamond\,(J = 0))$. To solve this, we first assign the minimal value $q_0(s)$ to each state such that there is a path originating in s which has at most $q_0(s)$ penalty in each step and reaches $(J = 0)$, i.e., $s \models \exists(\mathsf{penalty} \leq q_0(s)\,\mathrm{U}\,(J = 0))$, which can be calculated by a modified version of Dijkstra's algorithm in polynomial time. The series $(q_i(s))_{i\in\mathbb{N}}$ with $q_{i+1} = \min_{\alpha\in\mathrm{Act}(s)}\{\ \mathsf{penalty}(s,\alpha) + \max_{s',P(s,\alpha,s')>0} q_i(s')\ \}$ then converges to a value $q(s)$ which is the minimal value q in question. Note that in an acyclic MDP as given here, there is actually no iteration necessary and the value q can be calculated directly via back-propagation from the terminal states, i.e., the states with $(J = 0)$.

Further objectives based on the accumulation techniques in [3] can be found in Appendix B. Among others, the common index Power Usage Efficiency (PUE) can be expressed by them.

5 Experiments

We used the tool ProFeat [12] for specifying a parameterized version of the base model as described in Section 4. ProFeat is then used for creating the relevant instances for fixed parameter sets. Throughout this section we will report on three instances as shown in Table 1a.

With its parameters, $\mathcal{M}_{24}$ can be thought of as having a time resolution of one hour steps over one day. Similarly, $\mathcal{M}_{96}$ has a time resolution of 15 minutes. The capacity and the maximal production of green energy are fixed to 4 for all

Table 1: Considered instances of models, formulas and requests.

(a) Model and formula instances

	time T	jobs J_0	n in Form.(6)	n in Form.(8)
$\mathcal{M}_{24}$	24	60	40	6
$\mathcal{M}_{48}$	48	120	80	12
$\mathcal{M}_{96}$	96	240	160	24

(b) Requests

	shape	probability
R_0	no request	0.7
R_1	$([t, t+2], 0, 2)$	0.12
R_2	$([t, t+2], 2, 4)$	0.18

instances. We considered here two possible DR-requests with time-independent distribution as shown in Table 1b. The probability distributions for the service load are modeled in a time-independent fashion and are given by binomial distributions with a trail success probability of 0.4. The probability distributions for the load are time-dependent and pre-generated with random values. Table 2 shows the number of reachable states in the MDP for the three instances.

The model instances generated by ProFeat are in the input format of the prominent probabilistic model checker PRISM [21]. We used PRISM's symbolic engine, which uses multi-terminal binary decision diagrams (MTBDDs) for representing MDPs. As the size of MTBDDs crucially depends on the order in which variables occur in the MTBDD, we applied the reordering techniques described in [19] to end up with more compact model representations. This step was very effective, as the number of MTBDD nodes and hence the memory consumption could be reduced by up to 95%. This is reflected in Table 2 where the number of MTBDD nodes before and after reordering and the time for reordering is depicted. For the analysis we used the development version of PRISM 4.3 with additional implementations of the algorithms for computing conditional probabilities and quantiles. For the entire result section we used $\varepsilon = 10^{-4}$ (absolute values) for the convergence check of the numerical methods. Our experiments were run on a machine with an Intel Xeon E5-2680 CPU with 16 physical cores clocked at 2.7 GHz. The symbolic engine never exceeded the 1GB memory limit. The models together with the tools are available with the extended version [4] under `https://wwwtcs.inf.tu-dresden.de/ALGI/PUB/ATVA16/`.

5.1 Results

Table 2 shows the sizes of the generated models and the time for composing the model. It can be seen that our model scales well with increasing time resolution. A general overview of the model checking results is given in Table 3. It shows the result for each numbered non-quantile query in Section 4.3 and the time it took to process it. Again, we can see that the model checking times scale well with the time resolution of the model. Even for 96 time steps the model checking times are acceptable. Formulas 1-5 show that the basic confidence in our model is high. There are no deadlock states as can be seen in Formulas 1 and 2, i. e., the time always progresses to its final value. The probability to globally use green energy only is very low (Formula 4) and even lower when we are trying to finish all jobs (Formula 5), which is to be expected in this scenario. Similarly, the maximal probability for finishing all jobs in time (Formula 3) is smaller than 1

Table 2: Model sizes and build times.

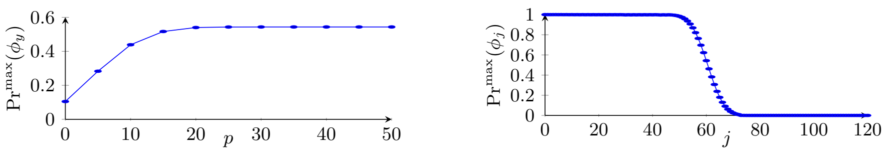
	reachable states	transitions	BDD nodes	reordered	reorder time	build time
$\mathcal{M}_{24}$	931,401	18,947,025	327,320	31,609	3.5 s	0.2 s
$\mathcal{M}_{48}$	3,841,426	81,269,400	551,716	39,203	6.9 s	0.3 s
$\mathcal{M}_{96}$	15,546,726	338,279,825	1,014,437	59,686	15.0 s	1.0 s

but positive which is expected under the given parameters. The value decreases with increasing model size due to the fixed capacity of 4 in each model instance even though the proportion between time and jobs stays the same.

The results for the Formulas 6-14 describe optimal values for single measures in the system. The probability for finishing two thirds of the total job pool (see Table 1a) is very high as can be seen in Formula 6. Accordingly, the expected number of finished jobs at the end of the day as seen in Formula 7 is close to the maximum. Formula 8 gives the probability to survive using at most 1/4 brown energy units on average per time step, which is almost impossible in the given setting. The expected total penalty until the end of the day is given in the Formulas 9 (maximal) and 10 (minimal). While the minimal penalty is low but non-zero, the maximal penalty is almost 1/2 units on average per two time step. The minimal penalty can be achieved by never scheduling any jobs. The expected total penalties until finishing all jobs is however infinite (see Formulas 11 and 12) since it is not guaranteed that all jobs will be finished (see Formula 3). Formulas 13 and 14 give the expected number of time steps until all jobs are finished or time runs out. The reason for these numbers being so close to the maximal number of time steps is that the job pool size is very close to the expected number of jobs to be finished (see Formula 7). Figure 3 shows quantile values for a variant of $\mathcal{M}_{24}$ with 120 initial jobs. Figure 3b shows the maximal probabilities for finishing a certain number of jobs until the end of the day. As expected, the probability is decreasing with higher job requirements and drops around the expected number of jobs calculated in Formula 7. The maximal probabilites for finishing all jobs with a given penalty bound is shown in Figure 3a. Probabilities of at least 0.6 cannot be reached, independent of

Table 3: PMC-results (see Section 5.1).

	24 steps / 60 jobs		48 steps / 120 jobs		96 steps / 240 jobs	
formula	result	time	result	time	result	time
(1)	true	0.1 s	true	0.0 s	true	0.1 s
(2)	true	0.4 s	true	0.2 s	true	0.7 s
(3)	0.543	31.5 s	0.392	104.2 s	0.244	748.4 s
(4)	$1 \cdot 10^{-4}$	1.7 s	$7 \cdot 10^{-5}$	2.4 s	$5 \cdot 10^{-5}$	6.8 s
(5)	0	3.8 s	0	0.9 s	0	2.3 s
(6)	0.999	25.9 s	0.999	128.9 s	0.999	1062.8 s
(7)	56.66	52.5 s	114.252	68.2 s	229.610	569.8 s
(8)	0.059	5.4 s	0.012	25.0 s	$6 \cdot 10^{-4}$	244.5 s
(9)	12.737	26.0 s	25.617	27.7 s	51.380	239.3 s
(10)	0.897	1.3 s	1.805	31.8 s	3.620	275.5 s
(11)	∞	0.0 s	∞	0.0 s	∞	0.0 s
(12)	∞	0.6 s	∞	1.3 s	∞	9.3 s
(13)	23.376	3.5 s	47.431	19.9 s	95.559	421.8 s
(14)	23.999	0.4 s	48.000	0.7 s	95.999	4.1 s
(18)	1.0	51.7 s	1.0	111.0 s	1.0	317.5 s
(19)	1.0	81.9 s	1.0	121.6 s	1.0	292.9 s



(a) $\phi_y = \Diamond^{\mathsf{penalty} \leq y}(J = 0)$ **(b)** $\phi_j = \Diamond^{\mathsf{\#jobs} \geq j}\ \mathsf{eod}$

Fig. 3: Probability values for ϕ_y and ϕ_j.

the penalty bound. This is immediately obvious from the results of Formula 3. The calculated quantile values are also useful for protocol design, since they give optimal parameters for jobs and penalties while retaining guarantees on the system reliability. A detailed overview of the quantile values for Formulas 15 and 17 can be found in [4]. The conditional probabilities for Formulas 18 and 19 are always 1. Intuitively, a maximizing scheduler for Formula 18 can try to work as many jobs as possible with green energy and can choose not to schedule any jobs anymore as soon as brown energy needs to be used. On the other hand, a scheduler for Formula 19 can start using brown energy as soon as the jobs cannot be finished anymore.

6 Conclusion

The purpose of the paper was to show the general feasibility of probabilistic model checking techniques for the analysis of demand-response systems. We provided a compositional model with components for the service load, the green-energy forecast, a request generator and an abstract model for the data center. Each components can easily be adapted and refined. We identified a series of important functional properties and performance measures that can serve as evaluation criteria for different strategies for scheduling jobs and provide useful insights for the design and refinement of the energy-aware workload management in data centers. The report on the experimental studies carried out with the model checker PRISM shows that several performance measures relevant to real systems are computable in reasonable time frames, up to a time resolution of 15 minutes. At the same time, this scenario can be seen as a stress test for the calculation of quantile values, in which the reordering techniques of [19] were crucial. Future work will include the consideration of variants of the model as discussed in Section 4 for different DR protocols with distributions that are derived from a real-world data center.

References

1. C. Baier, M. Daum, C. Dubslaff, J. Klein, and S. Klüppelholz. Energy-utility quantiles. In *Proc. of the 6th NASA Formal Methods Symposium (NFM)*, volume 8430 of *LNCS*, pages 285–299. Springer, 2014.
2. C. Baier, J. Klein, S. Klüppelholz, and S. Märcker. Computing conditional probabilities in Markovian models efficiently. In *Proc. of the 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 8413 of *LNCS*, pages 515–530. Springer, 2014.
3. C. Baier, J. Klein, S. Klüppelholz, and S. Wunderlich. Weight monitoring with linear temporal logic: Complexity and decidability. In *Proc. of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 11:1–11:10. ACM, 2014.
4. C. Baier, S. Klüppelholz, H. de Meer, F. Niedermeier, and S. Wunderlich. Greener bits: Formal analysis of demand response. Technical report, TU Dresden, 2016. See `http://wwwtcs.inf.tu-dresden.de/ALGI/PUB/ATVA16/`.
5. V. Balijepalli, V. Pradhan, S. Khaparde, and R. Shereef. Review of demand response under smart grid paradigm. In *Innovative Smart Grid Technologies-India (ISGT India), 2011 IEEE PES*, pages 236–243. IEEE, 2011.
6. R. Basmadjian, G. Lovasz, M. Beck, H. de Meer, X. Hesselbach-Serra, J. Botero, S. Klingert, M. P. Ortega, J. Lopez, A. Stam, R. van Krevelen, and M. D. Girolamo. A generic architecture for demand response: the ALL4Green approach. In *Proc. of the 3rd International Conference on Cloud and Green Computing (CGC)*, pages 464–471. IEEE, 2013.
7. R. Basmadjian, L. Mueller, and H. de Meer. Data centres' power profile selecting policies for demand response: Insights of green supply demand agreement. *Ad Hoc Networks*, 25:581–594, 2015.
8. J. L. Boudec and D.-C. Tomozei. Stability of a stochastic model for demand-response. *Stochastic Systems*, 3(1):11–37, 2013.
9. S. Chan and F. Schweppe. A generation reallocation and load shedding algorithm. *IEEE Trans. on Power Apparatus and Systems*, 1(PAS-98):26–34, 1979.
10. H. Chen, A. K. Coskun, and M. C. Caramanis. Real-time power control of data centers for providing regulation service. In *Proc. of the 52nd IEEE Conference on Decision and Control*, pages 4314–4321, 2013.
11. N. Chen, X. Ren, S. Ren, and A. Wierman. Greening multi-tenant data center demand response. *SIGMETRICS Perform. Eval. Rev.*, 43(2):36–38, 2015.
12. P. Chrszon, C. Dubslaff, S. Klüppelholz, and C. Baier. Family-based modeling and analysis for probabilistic systems - featuring ProFeat. In *Proc. of the 19th International Conference on Fundamental Approaches to Software Engineering (FASE)*, volume 9633 of *LNCS*, pages 287–304. Springer, 2016.
13. A. David, J. Grunnet, J. Jessen, K. G. Larsen, and J. I. Rasmussen. *9th International Symposium on Formal Methods for Components and Objects (FMCO). Revised Papers*, chapter Application of Model-Checking Technology to Controller Synthesis, pages 336–351. 2012.
14. V. Forejt, M. Z. Kwiatkowska, G. Norman, and D. Parker. Automated verification techniques for probabilistic systems. In *Proc. of the 11th International School on Formal Methods for the Design of Computer, Communication and Software Systems (SFM)*, volume 6659 of *LNCS*, pages 53–113. Springer, 2011.

15. A. Gandhi, V. Gupta, M. Harchol-Balter, and M. Kozuch. Optimality analysis of energy-performance trade-off for server farm management. *Performance Evaluation*, 67(11):1155–1171, 2010.
16. G. M. Huang and N. Nair. An OPF based algorithm to evaluate load curtailment incorporating voltage stability margin criterion. In *Proc. of NAPS conference*, 2001.
17. J.-P. Katoen. Model checking: One can do much more than you think! In *Proc. of the 4th IPM International Conference on Fundamentals of Software Engineering (FSEN)*, volume 7141 of *LNCS*, pages 1–14. Springer, 2011.
18. H. Kaur, Y. S. Brar, and J. S. Randhawa. Optimal power flow using power world simulator. In *Proc. of the 2010 IEEE Electric Power and Energy Conference (EPEC)*, pages 1–6, 2010.
19. J. Klein, C. Baier, P. Chrszon, M. Daum, C. Dubslaff, S. Klüppelholz, S. Märcker, and D. Müller. Advances in symbolic probabilistic model checking with PRISM. In *Proc. of the 22th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 9636 of *LNCS*, pages 349–366. Springer, 2016.
20. S. Klingert, F. Niedermeier, C. Dupont, G. Giuliani, T. Schulze, and H. de Meer. Renewable energy-aware data centre operations for smart cities the DC4Cities approach. In *Proc. of the International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)*, pages 1–9, 2015.
21. M. Z. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Proc. of the 23rd International Conference on Computer Aided Verification (CAV)*, volume 6806 of *LNCS*, pages 585–591, 2011.
22. Z. Liu, I. Liu, S. Low, and A. Wierman. Pricing data center demand response. In *Proc. of the 2014 ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, pages 111–123, 2014.
23. T. Medicherla, R. Billinton, and M. Sachdev. Generation rescheduling and load shedding to alleviate line overloads-analysis. *IEEE Trans. on Power Apparatus and Systems*, PAS-98(6):1876–1884, 1979.
24. H. Meidani and R. Ghanem. Multiscale Markov models with random transitions for energy demand management. *Energy and Buildings*, 61:267–274, 2013.
25. F. Niedermeier, W. Duschl, T. Möller, and H. de Meer. Increasing data centre renewable power share via intelligent smart city power control. In *Proc. of the 2015 ACM International Conference on Future Energy Systems*, pages 241–246, 2015.
26. G. Norman, D. Parker, M. Kwiatkowska, S. Shukla, and R. Gupta. Using probabilistic model checking for dynamic power management. *Formal Aspects of Computing*, 17(2):160–176, 2005.
27. M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, 1994.
28. A. Qureshi. *Power-demand routing in massive geo-distributed systems.* PhD thesis, Massachusetts Institute of Technology, 2010.
29. P. Siano. Demand response and smart grids—a survey. *Renewable and Sustainable Energy Reviews*, 30:461–478, 2014.
30. Z. Zhou, F. Liu, B. Li, B. Li, H. Jin, R. Zou, and Z. Liu. Fuel cell generation in geo-distributed cloud services: A quantitative study. In *Proc. of the 34th International Conference on Distributed Computing Systems (ICDCS)*, pages 52–61, 2014.
31. Z. Zhou, F. Liu, and Z. Li. Pricing bilateral electricity trade between smart grids and hybrid green datacenters. In *Proc. of the 2015 ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, pages 443–444. ACM, 2015.

A Additional Model Variants

We introduce another variant for the weather forecast component in this section.

Non-linear forecast. We can introduce a branching variant of the green energy forecast to achieve a more realistic model of its behavior. There, distributions for a time step can depend on the values of previous time steps. For example, we can express dependencies like "if it rains at noon, the probability for rain in the afternoon is high." Formally, we modify the random production variable to depend on the history, i. e., $\mathsf{produce}(t, v_0, ..., v_t) \in \{\ 0, 1, \ldots, \mathsf{prodmax}\ \}$ where $v_0, ..., v_t$ are the actual production values for the previous time steps.

B Accumulation Objectives

To have finer control over resource bounds, we can formulate accumulation objectives in the spirit of [3]. There, an operator $\oplus^{\text{filter}}(\text{constr})$ is given which forces the accumulated resources along any path fragment matching filter to match the constr. It allows to talk about the restricted accumulation of resources over a specific time frame or process. For example, a sliding window objective may be the requirement that over any t time steps there are always at least 5 jobs scheduled.

$$\Box \oplus^{\mathsf{\#steps} \leq t}(\mathsf{\#jobs} \geq 5)$$

These accumulation assertions can be embedded in more complex formulas, such as this one stating that whenever a request with time interval I arrives, not more than 5 brown energy units are used while it is active.

$$\Box(\mathsf{request}_I \rightarrow \oplus^{\mathsf{\#steps} \in I}(\mathsf{brown} \leq 5))$$

Accumulation objectives also allow the combination of resources in a constraint. For example, we can require the average penalty per finished job or per request taken to be smaller than c.

$$\oplus^{=T}\left(\frac{\mathsf{penalty}}{\mathsf{\#jobs}} < c\right) \qquad \oplus^{=T}\left(\frac{\mathsf{penalty}}{\mathsf{\#requests}} < c\right)$$

Similarly, guarantees on the Power Usage Efficiency can be expressed with additional reward functions energy and $\text{energy}_{\text{IT}}$ capturing the energy used in total and only by the IT equipment.

$$\oplus^{=T}\left(\frac{\text{energy}}{\text{energy}_{\text{IT}}} > c\right)$$

The following formula states that always more green energy than brown energy is used.

$$\oplus^{=T}(\mathsf{green} > \mathsf{brown})$$

C Additional Results

The concrete quantile values for Formulas 15 and 17 are given in Table 4b below. They correspond to the graphs seen in Figure 3

Table 4: Quantile values for ϕ_y and ϕ_j

(a)

p	0.1	0.2	0.3	0.4	0.5	0.6
y	0	3	6	9	14	∞

(b)

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
j	66	64	63	61	60	59	57	56	54	12