Multi-Agent Device-Level Modeling Framework for Demand Scheduling

Andreas Veit
Department of Computer Science
Cornell University
Ithaca, New York
Email: andreas@cs.cornell.edu

Hans-Arno Jacobsen
Department of Computer Science
Technische Universität München
Munich, Germany
Email: jacobsen@in.tum.de

Abstract—A problem receiving increasing attention is autonomously adjusting the electricity demand of consumers in response to real-time supply. The current literature mostly operates under the assumption that it is desirable to have an autonomous energy management system. However, autonomous demand-side management requires that smart agents understand their decision space. To model the decision space, it is key to model the scheduling constraints of individual devices under the agent’s control. Some constraints can be set by the owners, e.g., deadlines, while others are physical constraints. Recent work in demand-side management is based on a wide variety of models. In this work, we develop a standardized, device-level modeling framework characterizing the device categories by their constraints. Our models enable researchers and practitioners to develop and compare demand-side management algorithms and programs. Further, we evaluate the scheduling complexity of the different device categories and effects of different device combinations on the complexity. Our empirical results suggest that mixing different device categories can improve scheduling time.

I. INTRODUCTION

One key challenge in creating a sustainable society is to make consumer demand adaptive to the supply of electricity, especially to renewable supply. A problem receiving increasing attention is adjusting the demand of consumers via demand response programs (e.g., [1], [2]). While traditional demand response programs focus on large industrial customers [3], recent studies increasingly focus on household consumers as participants, (e.g., [4]). The current literature mostly operates under the assumption that it is desirable to have an autonomous agent that receives price signals and helps a consumer schedule its demand to minimize its electricity cost. [5] presents a tool that helps consumers model their preferences of their smart environment. Recent technological advances in smart meters and smart appliances have created the potential to enable direct and real-time participation of individual consumers in demand response programs. A few automated smart home research prototypes have been built, including KIT’s Smart Home [6] and MIT’s House n [7]. However, autonomous demand-side management requires that smart agents understand their decision space. To model the behavior of a device, it is fundamental to model the scheduling constraints of the device. For example, consider two populations, one fleet of electric vehicles (EVs) and one of residential HVAC systems. Both are attractive for demand response, because they consume a significant share of the total energy, but they differ in their consumption constraints. EVs have to ensure to complete charging before a set departure time, while HVACs constantly have to keep the temperature in a specific interval. Therefore, research in consumer coordination and demand scheduling requires models for the devices scheduled by the autonomous agents. However, recent work is based on a wide variety of device models with many authors developing their own models (cf. Section 2 for an overview). This diversity leads to a lack of comparison and restricted comparability of approaches as well as high entrance barriers to this area of research.

As main contribution, we develop a device-level modeling framework characterizing the different device categories by their individual constraints. Figure 1 gives an overview of the load categories in our framework. Our models enable comparisons between different solution approaches. To achieve comparability and interchangeability among different device categories, we use the Power Nodes Modeling Framework [8]...
as an interface. The framework models the interaction of devices with the grid, including efficiencies and power losses, but leaves out the specific constraints that define device behaviors. In this paper, we close this research gap by developing models for the device-specific constraints that fit into the Power Nodes Modeling Framework. To the best of our knowledge, there has been no framework combining all presented categories. The models have just enough detail so that key differences between the device categories can be expressed. Our framework can be used for demand response research, smart grid simulations or as reference model for the development of smart demand scheduling agents. In addition, we also contribute sample parameterizations for the devices that can be used for simulation-based evaluations. Finally, we empirically evaluate the scheduling complexity of the device categories and the effects of scheduling different types of devices together. Our results give insights into how mixing different device categories can improve scheduling efficiency.

II. RELATED WORK

Demand-side management and demand response receive increasing attention by research and industry. The approaches published so far include a variety of directions from direct load control to indirect incentive-based control (see [9] for an overview). In this work, we develop a device-level modeling framework for demand scheduling. We restrict this discussion to demand-side management research that builds models for the devices scheduled by the consumers in their algorithms.

The current literature on device models mostly focuses on shiftable loads that have to fulfill a certain task. The proposed models by [10], [11], [12], [13], [14], [15] resemble each other in that the loads are defined by their earliest start time, latest end time, a predefined energy necessary for the completion of the task and upper and lower bounds on the load in each time slot. We refer to these loads as continuous shiftable loads. [15] extends this basic model by allowing discrete load levels. However, none of these works consider non-interruptible devices that have to be serviced without interruptions. [13] motivates this omission by pointing out that scheduling these devices can be reduced to bin packing which is known to be an NP-hard combinatorial problem. In contrast to these papers, we not only model continuous and discrete shiftable loads, but also non-interruptible loads and show empirically how even non-interruptible devices can be scheduled in large groups.

Further literature considers the modeling of storage devices [12], [15], [16], [17]. These models have in common that the storage devices, in contrast to loads, do not have a specific task to complete, but provide storage capacity. Therefore, storage devices are modeled by constraints ensuring that the maximum and minimum storage capacity is complied with. However, none of these models takes into account charging and discharging efficiencies as well as storage losses. With our framework, we allow for these key parameters.

Further work considers the models of thermostatically controlled devices. These model have in common that they define thermal dynamics of devices. However, they vary in how detailed they reflect the physical foundation. Examples include [15], [18], [19], [20], [21]. In this work, we formulate the thermal dynamics similar to the Equivalent Thermal

Parameters approach, where the thermal dynamics are defined through simple parameters that can be fitted to the thermal characteristics of a device [20]. In contrast to the above mentioned models, we provide a generalized modeling framework that includes all categories of devices in one formulation.

III. GENERAL INTERFACE

As interface for the different device categories, we use the Power Nodes Modeling Framework [8]. This framework formalizes the interaction of devices with the grid. Let the set of all devices be denoted by \( D \) so that each device \( d \in D \). Further, let \( T \) denote the number of distinct time slots in a considered scheduling horizon so that each time slot \( t \in \{1, \ldots, T\} \). The dynamics of a device \( d \) in time slot \( t \) are formally defined by the following.

\[
C_d \cdot e_{st,d} = C_d \cdot e_{s,t-1,d} + \eta_{t,d} \cdot P_{t,d} - (\eta_{d,gen} \cdot P_{t,d} - x_{t,d} - v_{t,d}) \quad (1)
\]

s.t. (a) \( 0 \leq e_{st,d} \leq 1 \),
(b) \( 0 \leq \min_{t,d} \leq P_{t,d} \leq \max_{t,d} \),
(c) \( 0 \leq \min_{t,d} \leq P_{t,d} \leq \max_{t,d} \),
(d) \( 0 \leq v_{t,d} \quad \forall t \in \{1, \ldots, T\}, d \in D \).

On the left hand side, the storage capacity is denoted by \( C_d \). The current energy state of the device in timeslot \( t \) is \( e_{st,d} \), which is normalized between 0 and 1. The energy state in time \( t \) is given by the state in time \( t-1 \) plus the load drawn from the grid and minus the power fed into the grid, the power used for the device’s task and the storage loss. Constraint (a) ensures that the energy state remains within the storage limits. The load drawn from the grid is \( P_{t,d} \) and the respective efficiency is \( \eta_{t,d} \). Similarly, the power fed into the grid is \( P_{t,d} \) and the generation efficiency is \( \eta_{d,gen} \). The constraints (b) and (c) ensure that the load drawn from the grid and the power fed into the grid adhere to the lower and upper limits imposed by the connection to the grid. The external process that defines the device’s specific behavior is denoted as \( x_{t,d} \). In contrast to the formalization of Heussen et al. [8], for consistency with the standard notation for device-level modeling, we define a positive external process to be a consumption of energy. Lastly, the storage loss is denoted as \( v_{t,d} \). Constraint (d) ensures that storage loss leads to a reduction of the energy state. The efficiency and loss terms can also be modeled to be dependent on the current state of the unit, i.e., \( \eta_{t,d}^{load}(e_{st,d}) \), \( \eta_{t,d}^{gen}(e_{st,d}) \) and \( v_{t,d}(e_{st,d}) \). In summary, the constraints (a) - (d) are linear and form a convex set. For readability, we omit the subscript \( d \) indicating the specific device in the remainder of this paper.

IV. DEVICE MODELS

Devices can generally be divided into two groups: First, loads consuming a predetermined amount of energy to fulfill a certain task. Second, loads that offer storage capacity and have to ensure that the energy state stays within energy levels. The first group can be divided into non-shiftable loads that have to run at fixed times and shiftable loads like electric vehicles or dishwashers that have to complete their task before a specified deadline. Shiftable loads can further be divided into loads with continuous or discrete load levels and non-interruptible loads that, once started, have to be serviced without interruption.
The second group can be divided into electric storage devices and thermostatically controlled loads (thermal devices) like refrigerators or HVAC systems. See Figure 1 for an overview. We now formally define the different device categories.

**Definition 1.** A load is a device that has to fulfill a specific task and has a predefined total energy consumption required to fulfill that task.

With the set of all time slots $[1, ..., T]$, the decision variables of a load are its power consumption in each time slot $x = [x_1, ..., x_T]$. Since loads only consume energy and do not supply to the grid, $d^{	ext{cen}}_t = 0$. Further, loads offer no storage capabilities, i.e., $C = 0$, and thus experience no storage loss, i.e., $v_t = 0$. Devices that are loads but also contain a buffer can be modeled by a load plus a storage device. For loads, Equation 1 is reduced to

$$
  x_t = \eta_t^{\text{load}} p^L_t
$$

with constraint (b) it follows that $x_{t,d} \geq 0$.

**Definition 2.** A non-shiftable load is a device that has a predefined power consumption profile for a given time interval.

Let the device’s predefined consumption profile that must be satisfied in each time slot be $r \in \mathbb{R}^T$ so that

$$
  x_t = r_t, \forall t \in [1, ..., T]
$$

A non-shiftable load can be modeled by its profile $r$ s.t. (b), (2) and (3). Examples include entertainment and office appliances that need to run exactly when required. Consumption of these devices has strict constraints so that no scheduling is possible.

**Definition 3.** A shiftable load is a device that has a predefined total power consumption for a given time interval.

Shiftable loads can defer or sometimes even interrupt their task, as long as they complete it in time. Let $E$ denote the total energy consumption required for the device’s task. In addition, loads might not be available during the whole scheduling horizon, e.g., washing machines can only run after they have been loaded and might have to be finished at a certain time. Let $l$ denote the earliest start time and $\bar{t}$ the latest end time of a device. Then, the first constraint ensures that, summed over the available time slots, the device’s consumption provides enough energy for the device’s task. The second constraint ensures that the device does not consume energy while not available.

$$
  \sum_{t=l}^{\bar{t}} x_t \geq E; \quad x_t = 0, \forall t \notin [l, ..., \bar{t}]
$$

To allow efficiencies dependent on the current state of the device, we model the state of a load as

$$
  e_{st} = \sum_{i=l}^{t} x_i / E
$$

where $\sum_{i=l}^{t} x_i$ is the energy already consumed up to time slot $t$ and $E$ is the total required energy. Shiftable loads can further be divided into the following three subcategories.

**Definition 4.** A continuous load is a shiftable load that can scale its power levels within a continuous interval.

The consumption of continuous loads in each time slot, where the device is plugged in, is constrained by a minimum standby level, $L_t^{\text{min}}$, and a maximum power level, $L_t^{\text{max}}$, i.e.,

$$
  0 \leq L_t^{\text{min}} \leq x_t \leq L_t^{\text{max}}, \forall t \in [l, ..., \bar{t}]
$$

A continuous load can be modeled by the parameters $E$, $L_t^{\text{min}}$, $L_t^{\text{max}}$, $l$ and $\bar{t}$ s.t. (b), (2), (4) and (6) and, in case of state-dependent efficiencies, with the addition of (5). These constraints form a convex and differentiable scheduling problem. Table I gives a parametrization of a simplified EV.

**Definition 5.** A discrete load is a shiftable load that has a set of discrete power levels.

The power consumption of discrete loads can only be set to predefined power levels defined by the set $L$, e.g., on or off.

$$
  x_t \in L = \{L_1, ..., L_n\}, 0 \leq L_1, ..., L_n, \forall t \in [l, ..., \bar{t}]
$$

A discrete load can be modeled by the parameters $E$, $L$, $l$ and $\bar{t}$ s.t. (b), (2), (4) and (7) and, in case of state dependent efficiencies, with the addition of (5). These constraints form a convex and linear combinatorial scheduling problem. Table I gives a parametrization for an EV with discrete load levels.

**Definition 6.** A non-interruptible load is a shiftable load that, once started, has to follow a predefined consumption profile.

Non-interruptible loads combine characteristics from both non-shiftable and shiftable loads. The start time is flexible and the load can be shifted, but once started the load has to follow a predefined consumption profile, $r$, and cannot be interrupted. Thus, the scheduler can only set the starting time of the device.

Let $\text{len}$ denote the duration of the device’s task so that $r \in \mathbb{R}^{\text{len}}$ is its consumption profile. The starting time of a device is modeled by the binary variable $o_{nt}$, that takes the value $o_{nt} = 1$ in the time slot the load is started and zero in all other time slots. Equation 8 ensures that, once turned on in time slot $t$, the consumption $[x_{t}, ..., x_{t + \text{len}}]$ follows the required profile $r$. Equation 9 ensures that the device is only turned on once in the planning horizon.

$$
  [x_{t}, ..., x_{t + \text{len}}] \geq r \cdot o_{nt}, \forall t \in [l, ..., \bar{t} - \text{len}], o_{nt} \in \{0, 1\}
$$

$$
  \sum_{t=1}^{\bar{t}} o_{nt} = 1; \quad \sum_{t=1}^{\bar{t}} x_t = \sum \text{r}
$$

A non-interruptible load can be modeled by the parameters $r$, $l$ and $\bar{t}$ s.t. (b), (8) and (9). These constraints can be reduced to a bin-packing problem [13], which is known to be an NP-hard combinatorial problem. Table I gives exemplary parameterizations for a washer/dryer and a dishwasher. Although in practice, these devices can be paused to some degree. However, in industry settings many processes are non-interruptible due to physical and chemical constraints.

**Definition 7.** A storage is a bi-directional device that offers storage capacity so that it can take in as well as provide energy, depending on whether it is charging or discharging.

\[E^{\text{V}}\]s have been modeled as continuous shiftable loads for simplicity of analysis and to develop efficient scheduling algorithms, e.g., [11], [13]. However, in commercial EVs, the charging rates are discrete.
The decision variables of a storage are its charging and discharging levels in each time slot, i.e., $p_{t}^{load}, p_{t}^{gen} \geq 0$. Storages do not perform a specific task that consumes energy, i.e., $x, E = 0$. Thus, Equation 1 gets reduced to

$$Ces_{t} = Ces_{t-1} + \eta_{t}^{load} p_{t}^{load} - \eta_{t}^{gen} p_{t}^{gen} - v_{t}$$

(10)

Some storage devices might have separate conversion units for charging and discharging so that both conversions can happen at the same time, while for others one of the variables must always be zero. Let $sep_{conv} \in \{0,1\}$ denote whether a storage can charge and discharge at the same time. Equation 11 ensures that if $sep_{conv} = 0$, either the load $p_{t}^{load}$ or the generation $p_{t}^{gen}$ has to be 0 as well in each time slot.

$$p_{t}^{load} \cdot (1 - sep_{conv}) = 0, \text{ } sep_{conv} \in \{0,1\}$$

(11)

A storage can be modeled by the parameters $C, \eta_{t}^{load}, \eta_{t}^{gen}, v_{t}$ and $sep_{conv}$ s.t. (a), (b), (c), (d), (6 or 7), (10) and (11). With separate conversion units, these constraints form a convex and linear scheduling problem that can be either continuous or discrete depending on the constraints on the load levels. Without separate conversion units, the scheduling problem becomes quadratically constrained. Table I gives an exemplary parametrization for a battery, usually installed together with residential solar PV modules.

**Definition 8.** A thermal load is a device that consists of a heat store with a system temperature that must be kept within temperature limits.

Thermal loads also offer storage capacity, i.e., $C > 0$. However, what distinguishes them from storage devices is that they are typically uni-directional and cannot provide energy, i.e., $p_{t}^{gen} = 0$. Thus, Equation 1 gets reduced to

$$Ces_{t} = Ces_{t-1} + \eta_{t}^{load} p_{t}^{load} - v_{t}$$

(12)

Further, thermal devices are driven by their thermal dynamics. Let $\theta_{amb}$ be the ambient temperature around the thermal device and $\theta_{i}$ be the system temperature inside. Without the loss of generality, we assume a heating device, i.e., the ambient temperature is lower than the system temperature. In contrast to storage devices, the storage capacity is typically not physically constrained, but by user-defined lower and upper bounds on the system temperature, i.e., $\theta_{min}$ and $\theta_{max}$. All temperatures are in °C. Let the device’s heat capacity be $c$ measured in [MWh/°C]. Then, the storage capacity is

$$C = (\theta_{max} - \theta_{min}) \cdot c$$

(13)

Further, let the ambient conduction coefficient be $\kappa_{d}$ measured in [MW/°C]. The heat loss is defined based on Fourier’s law as $\theta_{loss} = \frac{\kappa_{d}}{c} (\theta_{i} - \theta_{amb})$. By expressing the system temperature in terms of the device’s energy state, we can express the storage loss as

$$v_{t}(es_{t}) = \kappa \left( \theta_{min} + \frac{Ces_{t}}{c} - \theta_{amb} \right)$$

(14)

A thermal device can be modeled by the parameters $\theta_{amb}, \theta_{min}, \theta_{max}, \eta_{t}^{load}, c$ and $\kappa$ s.t. (a), (b), (6 or 7), (12), (13), and (14). These constraints result in a recursive and discrete problem, because the load levels of thermal devices are typically discrete. Table I gives an exemplary parametrization of a residential water heater.

<table>
<thead>
<tr>
<th>Table I. Exemplary Parametrizations for Device Categories.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous load:</td>
</tr>
<tr>
<td>simplified EV</td>
</tr>
<tr>
<td>total energy $E = 5.14kWh$</td>
</tr>
<tr>
<td>min standby power $L_{min} = 0kW$</td>
</tr>
<tr>
<td>max power level $L_{max} = 7.6kW$</td>
</tr>
<tr>
<td>start/end times $\tau - \tau = 4.91h$</td>
</tr>
<tr>
<td>Discrete load:</td>
</tr>
<tr>
<td>electric vehicle $C = 5.7kWh$</td>
</tr>
<tr>
<td>power levels $L = (0, 7.6)kW$</td>
</tr>
<tr>
<td>Non-interruptable load:</td>
</tr>
<tr>
<td>washer/dryer $C = 418min$</td>
</tr>
<tr>
<td>storage $v_{t} = NA$</td>
</tr>
<tr>
<td>Thermal load:</td>
</tr>
<tr>
<td>water heater $C = 16mWh$</td>
</tr>
<tr>
<td>ambient temp $\theta_{amb} = 20°$</td>
</tr>
<tr>
<td>min temp system $\theta_{min} = 60°$</td>
</tr>
<tr>
<td>max temp system $\theta_{max} = 65°$</td>
</tr>
<tr>
<td>load efficiency $\eta_{load} = 100%$</td>
</tr>
<tr>
<td>heat capacity $c = 0.3487kWh/°C$</td>
</tr>
<tr>
<td>ambient cond. coeff. $\kappa = 0.00132kWh/°C$</td>
</tr>
<tr>
<td>power levels $L = (0, 3)kW$</td>
</tr>
</tbody>
</table>

V. EMPIRICAL EVALUATION OF SCHEDULING COMPLEXITY

We empirically evaluate the scheduling complexity of the device categories modeled in our framework. In particular, we evaluate how many devices can be scheduled together and how different compositions of devices affect the complexity. Demand response programs require different response latencies. In the balancing market, power needs to be drawn or fed 15 minutes after being dispatched. This gives 10 minutes to schedule devices and 5 minutes to ramp up. We simulate scheduling tasks for different device populations and measure the required time for scheduling. We also investigate the effect of mixed groups on scheduling complexity.

A. Objective Function

As objective we choose to fit the aggregated load of all devices to a predefined load curve, profile. In particular, the objective is to minimize the distance of the aggregated load of all devices, i.e., $\sum_{i \in devices} (load_{i,j} - gen_{i,j})$, to that curve. In our experiments, we use a flat profile.

$$\min C(load, gen) = \min \sum_{j \in timeslots} \lvert profile_{j} - \sum_{i \in devices} (load_{i,j} - gen_{i,j}) \rvert$$

(15)

2We use the 2014 BMW i3 model, which has an energy consumption of 12.9 kWh/100 km. We assume a 40 km drive and a standing time of 5 hours.
3We use the parameters of the Siemens WD14H420GB.
4We use the parameters of the Sonnenbatterie S.
5We use the parameters of the Feuron Variotherm FLEV with 300 liter water at 60°C and surface area 4m², insulation thickness 7mm and thermal conductivity 0.0233W/m·K.
The experiment was setup using Gurobi 5.6.2 with Python 2.7.6 on a Mac Book Pro 9,2 with an Intel Core i7 3520M with two cores operating at 2.9GHz and 8 Gigabyte of DDR 3 physical memory at 798Mhz running Windows 8.1. In practice, more powerful many-core systems could be used to handle larger groups of devices. However, this would not reduce the hardness of the scheduling problems.

D. Results

For our empirical evaluation we perform a total of 3645 schedule optimizations. We observe that, except for thermal devices, all device categories can be scheduled for population sizes up to several thousand devices. Figure 3 illustrates the runtime of the optimization for the different device categories up to 5,000 devices each. The data points for continuous and discrete devices are located close to the x-axis. For non-interruptible and storage devices, we observe that population sizes up to 3,000 and 4,000 devices are solvable within the set time limit of 10 minutes. However, thermal devices are only schedulable for up to 15-20 devices. The respective data points are located close to the y-axis. This is in line with the results from Boseman et al. [22], who were able to compute schedules for fleets of up to 10 combined heat and power generators within the limit of 10 minutes. For mixed groups, we observe that up to 2,000 devices can be scheduled within the time limit. This is a surprising result, because the group includes 400 thermal devices. This indicates that thermal devices benefit from being scheduled together with other categories. This is supported by the results of the following experiment.

Figure 4 illustrates results from groups with randomly drawn combinations of devices. The figure shows the 8th-order polynomial regression with standard error of the runtime over the different fractions of the individual device category in the mixed groups. Over all, we observe an average runtime of 231 seconds. For continuous devices, we do not see an effect of its share on the runtime of the optimization. For discrete devices, we see an increased runtime for fractions between 50% and 80% with a maximum around 65%. Non-interruptible devices show a similar pattern to discrete devices. Groups of only non-interruptible devices are scheduled efficiently, but combining them with other categories makes the scheduling problem significantly harder. In contrast, storage loads generally decrease runtime. In particular, groups with more than 20% storages experience shorter runtimes. One explanation could be that given a sufficient share, the storages can compensate the remaining loads. Lastly, for thermal devices, we observe...
a surprising pattern. The very long runtime towards fractions close to 100% is in line with our scalability result, i.e., optimal schedules for groups of only thermal devices are not solvable at a scale of 1,000 devices. However, for the interval between 25% and 75%, thermal devices seem to benefit strongly from being scheduled together with devices from other categories.

VI. CONCLUSIONS

In this paper, we developed formal models for shiftable loads with both continuous and discrete load levels, non-interruptible loads as well as storage and thermal loads. For each model, we developed a mathematical formulation and provided the key characteristics for the resulting schedule optimization problem, e.g., convexity or discreteness. Our models can help researchers as well as practitioners to develop new or choose the right solution concepts for their algorithms and programs. Further, we made the models interchangeable and comparable by using a common interface and provided sample parameterizations for the device categories to allow comparable simulations and empirical evaluations. Moreover, we evaluated the scheduling complexity of the different device categories as well as mixed groups with an empirical study. The evaluation showed that continuous and discrete shiftable loads can be scheduled up to very large groups, as well as that thermal devices show very limited scalability. Further, the evaluation gives an insight in how different compositions of mixed groups can affect the runtime of the schedule optimization. In particular, we show that thermal devices can be scheduled in large populations, if they are mixed with a sufficient number of different devices. These results are very promising as they get us closer to large scale autonomous demand response programs.

REFERENCES


