Abstract

Demand response (DR) for residential and small commercial buildings is estimated to account for as much as 65% of the total energy savings potential of DR, and previous work shows that a fully automated Energy Management System (EMS) is a necessary prerequisite to DR in these areas. In this paper, we propose a novel EMS formulation for DR problems in these sectors. Specifically, we formulate a fully automated EMS’s rescheduling problem as a reinforcement learning (RL) problem (referred to as the device based RL problem), and show that this RL problem decomposes over devices under reasonable assumptions. Compared with existent formulations, our new formulation (1) does not require explicitly modeling the user’s dissatisfaction on job rescheduling, (2) enables the EMS to self-initiate jobs, (3) allows the user to initiate more flexible requests and (4) has a computational complexity linear in the number of devices. We also propose several new performance metrics for RL algorithms applied to the device based RL problem, and demonstrate the simulation results of applying Q-learning, one of the most popular and classical RL algorithms, to a representative example.

Keywords: Demand Response, Energy Management System, Building and Home Automation, Reinforcement Learning, Markov Decision Process

1 Introduction

Demand response (DR) systems [Borenstein et al., 2002, Braithwait and Eakin, 2002, Barbose et al., 2004] dynamically adjust electrical demand in response to changing electrical energy prices or other grid signals. DR offers several benefits. By suitably adjusting energy prices, load can be shifted from peak energy consumption periods to other times. This, in turn, can improve operational efficiency, reduce operating costs, improve capital efficiency, and reduce harmful emissions and risk of outages. The variability of renewables can create an additional need to shift energy consumption in order to better match energy demand with unforecasted changes in electrical energy generation. The benefit is a reduction in backup (ancillary) generation frequently used to hedge renewable sources.

There are several types of DR. In direct DR a utility or other entity directly modifies the energy consumption of users by adjusting the operation of user’s equipment. Interruptible tariffs allow a utility to interrupt the supply of power to a company under predefined conditions. Price driven DR uses pricing mechanisms to attempt to modulate energy demand. DR has been extensively investigated for larger energy users and has been implemented in many areas (e.g., [Roos and Lane, 1998, Piete et al., 2005]). Residential and small commercial building DR [Herter, 2007b,a, Faruqui and George, 2005, Koch and Piete, 2007] offers similar potential benefits. DR for residential and
small commercial buildings was estimated to account for as much as 65% of the total energy savings potential of DR. However, DR in the residential and small commercial building sector faces several challenges.

Technical challenges include deploying an infrastructure supplying real-time pricing information to energy consumers\(^1\) in a useful way, ensuring security, and implementing advanced metering and networking devices [Koch and Piette, 2007, LeMay et al., 2008]. In addition to all these technical challenges, another challenge vital to the success of DR in the residential and small commercial building sectors is that it requires a fully automated Energy Management System (EMS) [Koch and Piette, 2007, Piette et al., 2007]. This is because with price driven DR, consumers face a continuing sequence of decisions to either use a particular device now and consume energy at current (known) prices or to defer using the device until later at possibly unknown prices. Each decision requires the consumer to weigh the cost differential against his dissatisfaction due to rescheduling device usage. This is particularly burdensome when the consumer must also estimate future energy prices. Further many of these decisions have limited financial impact on the consumer [ONEill et al., 2010], and, as a result, many rational consumers in the residential and small commercial building sectors may not be sufficiently incentivized to make these decisions over the long run (known as “decision fatigue” in [ONEill et al., 2010]).

To be effective, an EMS system needs to automatically make energy consumption decisions that are consistent with the cost delay trade-offs of energy users, in this way acting as an energy agent. It is often difficult to cost-effectively model the behaviors of the idiosyncratic consumers and the temporal variations of the energy prices, a successful DR EMS needs to learn to make optimal decisions for consumers from interacting with the consumers and energy prices. Recently, [ONEill et al., 2010] proposed a fully-automated EMS algorithm based on reinforcement learning (RL), which learns how to make optimal decisions for consumers. To the best of our knowledge, this is the first paper to apply RL to DR in residential and small commercial building sector. The authors adopt a request inventory model for the system dynamics and use Q-learning, a classical RL algorithm, to learn how to make the optimal decisions for energy consumers. In this approach, users make energy requests to the EMS system (e.g. pushing a button on a device that a user wants to run) and the system schedules the time of operation by calculating the user’s trade-offs between delay and energy prices. It learns these trade-offs by observing energy consumer’s behaviors and observing the patterns of energy pricing. Over time the EMS learns to make the best decisions for energy users in the sense that it balances energy cost and the delay in energy usage in the same way that the customer optimally would, but without the consumer having to make the decision. The authors explicitly assume that (1) consumers’ dissatisfaction with delay can be modeled by known disutility functions, and (2) that consumers explicitly initiate all energy usage. This approach has several limitations:

- Finding specific disutility functions for a particular residence or small business can be difficult and costly. In [ONEill et al., 2010], the authors assume these functions have particular mathematical properties, but do not address how these functions might be determined. These functions are likely to be idiosyncratic and are specific to energy price vs. time delay trade-offs.

- Many energy consuming activities occur without the consumer directly initiating them. HVAC in office buildings and pool heaters in residential settings are obvious examples. A useful EMS would self-initiate jobs for these and similar devices without explicit user requests or reservations. For example, if it is unexpectedly hot in a summer afternoon and the current energy

\(^1\)Throughout this paper, we use the terms “user” and “consumer” interchangeably.
price is expected to be cheaper than that in the evening, then the EMS should be allowed to turn on AC without an explicit request/reservation from the consumer.

- The computational complexity of this approach grows exponentially as the number of devices. Known as the “curse-of-dimensionality” in dynamic programming (DP) and RL literature, this problem limits the approach to fairly small numbers of devices.

In this paper, we propose a novel EMS formulation that addresses the limitations of [ONeill et al., 2010] described above. Our proposed algorithm also uses RL, but adopts a device centric point of view, and, as we will discuss in detail in Section 3, under reasonable assumptions, the RL problem decomposes over devices and it is sufficient to apply an RL algorithm to each individual device. Specifically, the algorithm addresses these issues:

1. Our EMS formulation does not require a pre-specified disutility function modeling the consumer’s dissatisfaction on job rescheduling. Instead, under this formulation, the EMS is able to learn the consumer’s dissatisfaction by sampling his evaluations on completed/canceled jobs. In other words, our new RL formulation has eliminated the impractical assumption in [ONeill et al., 2010] that consumers’ dissatisfactions with delay can be captured by known disutility functions.

2. Our approach allows both user-initiated jobs and EMS-initiated jobs. The EMS-initiated jobs use a probing/feedback mechanism to find the best way to anticipate future energy usage.

3. Our EMS algorithm also enables more flexible user-initiated jobs, specifically

   - A consumer request’s target time can be different from its request time, where the request time is the time when the EMS receives this consumer request, and the target time is the time when the consumer prefers this request to be satisfied.
   - Consumer requests/reservations can have different priorities, whereas in [ONeill et al., 2010], all the consumer requests have the same priority.
   - Energy requests can be canceled by the consumer, reflecting the behavior of real energy users.

4. The computational complexity of our approach grows linearly with the number of devices, and thus many classical RL algorithms can be applied even when there are a large number of devices.

In this paper, we also propose several new performance metrics for any RL algorithm applied to this problem. In particular, we suggest methods of measuring performance relative to the user’s current pattern of behavior and relative to a prescient optimal pattern of behavior.

The remainder of this paper is organized as follows. In Section 2, we briefly describe how a practical fully-automated EMS should interact with the consumer and the grid signals. In Section 3, we pose the optimal demand response problem faced by an EMS as an infinite horizon discounted Markov decision process (MDP), and decompose this high-dimensional MDP into a collection of low-dimensional device based MDPs under appropriate assumptions. Then, in Section 4, we propose the device based RL problem and discuss several performance metrics for RL algorithms applied to this problem. In Section 5, we demonstrate the simulation results on a representative example when the classical Q-learning is applied. We conclude in Section 6.
2 Description of fully-automated EMS

A fully-automated EMS (henceforth referred to as EMS) is a necessary prerequisite to DR in the residential and small commercial building sectors. Furthermore, an EMS needs to learn how to make the optimal decisions for the user while interacting with them and the real-time grid signals. In this section, we describe how a fully-automated EMS should interact with the user and the grid signals.

Generally speaking, a fully-automated EMS observes the grid signals, receives requests and evaluations from the user, and schedules the jobs over the devices (Figure 1). We detail the interaction mechanisms in the remainder of this subsection.

Grid Signals

The EMS observes the grid signals through a communication network, where the “communication network” refers to the infrastructure that supplies grid signals to the EMS. Any exogenous information that is effectively delivered by the communication network and is useful for the EMS to make the scheduling decisions can be regarded as a grid signal. The most common grid signal is the real-time energy price; other grid signals might include the expected future energy prices, the real-time temperature and weather condition, and other useful exogenous information.

In this paper, we assume that (1) the only available grid signal is the real-time energy price, and (2) the energy price is exogenous in the sense that the EMS’s actions will not influence it (i.e. the EMS is a price-taker). Notice that the second assumption is reasonable since in the residential and small commercial building sectors, the market power of an individual EMS (or equivalently, of an individual consumer) is so small that the impact of its actions on energy prices is negligible. On the other hand, the first assumption is proposed mainly to simplify the exposition and does not incur loss of generality. Specifically, both the model and the algorithm proposed in this paper can be applied to the cases with more general grid signals (e.g. the case in which the grid signals are energy price and temperature), as long as the grid signals are exogenous.

User-Initiated Jobs and EMS-Initiated Jobs

A fully-automated EMS should perform the following functions:
The EMS receives requests from the consumers, and then schedules when to fulfill the received requests. We henceforth refer to this case as a *user-initiated job*.

If a smart device managed by the EMS is idle (i.e. currently there is no request for that device), the EMS could speculatively power that device. We henceforth refer to this case as an *EMS-initiated job*. For instance, in a small commercial building, the EMS might speculatively turn on the building’s air conditioning in advance of the tenant’s arrival to capture early morning lower energy costs or to mask the latency of cooling the building. Notice that we should not allow the EMS to do speculative jobs on all the smart devices (such as dishwashers).

We assume time is discrete \( t = 0, 1, \cdots \) and that there are \( N \) smart devices managed by the EMS and numbered \( n = 1, 2, \cdots , N \). To simplify exposition, we assume that all the jobs done by device \( n \) are standardized and hence they can be completed in one time step and consume a constant energy \( C_n \), which only depends on the type of the smart device. This assumption can be readily relaxed to devices with different operating periods. We further assume that the EMS will ignore all the user requests to device \( n \) if device \( n \) currently has an uncompleted request, but the EMS allows a consumer to cancel an existing uncompleted request. Specifically, if the consumer wants to replace an existing request with a new request, he must cancel the existent request first, and then start the new request. Notice under this assumption, at each time a smart device has at most one uncompleted request.

**Interaction between user and EMS**

We now describe how the EMS interacts with the consumer (user) in a user-initiated job and an EMS-initiated job. Specifically, as its name suggests, a user-initiated job starts with a user sending a request to the EMS. Specifically, each consumer request is represented by a four-tuple \( J = (n, \tau_r, \tau_g, g) \), where

- \( n \) denotes the requested device;
- \( \tau_r \) is the *request time* and denotes when the EMS receives this request;
- \( \tau_g \) is the *target time* and denotes when the user prefers this requested job to be completed;
- \( g \) denotes the *priority* of this request, with higher priority implying the “stronger preference” of the user that they want the requested job to be completed at a time close to the target time \( \tau_g \).

Notice that the target time \( \tau_g \) is not necessarily equal to the request time \( \tau_r \); instead, the user might request to use a device in a later time (i.e. \( \tau_g \geq \tau_r \)). On the other hand, for a request \( J = (n, \tau_r, \tau_g, g) \), it is unreasonable to assume that \( \tau_g - \tau_r \), the difference between the target time and the request time, can be arbitrarily large. Thus, in this paper, we assume that (1) for any request \( J = (n, \tau_r, \tau_g, g) \), its target time \( \tau_g \) must satisfy \( \tau_r \leq \tau_g \leq \tau_r + W_n \), and (2) if the request \( J = (n, \tau_r, \tau_g, g) \) is not fulfilled by time \( \tau_g + W_n \), then it will be canceled by the user, where \( W_n \) is a known time window and only depends on the type of device.

We also assume that an unsatisfied consumer request can be canceled by the user. Furthermore, we assume that when a consumer request is completed or canceled, the user will send an evaluation on the completed/canceled request to the EMS. As we will see later, the EMS can use such evaluations to learn the user’s dissatisfaction on the rescheduling of the user-initiated jobs.

On the other hand, an EMS-initiated job is started by the EMS, without receiving a request from the user. The only interaction between the EMS and the user for such jobs is that once an
EMS-initiated job is completed, the user will send an evaluation on it to the EMS. As is in the user-initiated jobs, the EMS also exploits such evaluations to learn user’s dissatisfaction with the EMS-initiated jobs.

In summary, the interaction between user and EMS is as follows: for user-initiated jobs, the possible interactions include (1) the user sends requests to the EMS, (2) the user can choose to cancel the unsatisfied requests and (3) the user evaluates the completed/canceled requests. On the other hand, for EMS-initiated jobs, the only interaction between user and EMS is that the user evaluates the completed EMS-initiated jobs.

To complete the description of the user-EMS interaction, we need to specify the “user-EMS interaction timeline” in a single time period (i.e. whether the EMS receives the user requests/cancellations/evaluations before or after it schedules jobs in that time period). Without loss of generality, in this paper, we consider the user-EMS interaction timeline illustrated in Figure 2.

![Figure 2: User-EMS interaction timeline in time period t](image)

3 Device Based DR Model

This section proceeds as follows. We first motivate and define the dissatisfaction function and the instantaneous cost function for a rational consumer in Subsection 3.1 and 3.2, respectively. Explicit dissatisfaction functions are not required in practice, but in this section we assume they are known to facilitate easy exposition. Then, Subsection 3.3 formulates the optimal demand response problem as a collection of device-based MDPs under suitable assumptions.

3.1 Dissatisfaction Function and Rational Consumers

To formalize the notion of optimal demand response, in this subsection, we define the dissatisfaction function for a consumer (user) that captures their preferences (dissatisfaction) over job rescheduling. We also motivate and propose a specific class of functional forms for the dissatisfaction functions of the rational consumers.

Let us start by defining some useful notation. Specifically, for each time $t$, we define the sets of smart devices $D(t), C(t) \subseteq \{1, 2, \cdots, N\}$ as

$$\begin{align*}
D(t) &= \{\text{devices that do a job at time } t\} \\
C(t) &= \{\text{devices for which an unsatisfied request is canceled at time } t\}
\end{align*}$$

Note that $D(t) \cap C(t) = \emptyset$, since if device $n$ does a job in time period $t$, then there is no unsatisfied request for device $n$ at the end of time period $t$ (recall that at any time, device $n$ has at most one unsatisfied request, and see Figure 2 for the timeline of time period $t$). Furthermore, we use $J(t)$ to denote the set of unsatisfied requests right after the requests in time period $t$ have been
received. Since at any time device $n$ has at most one unsatisfied request, thus, if $J(t)$ contains one unsatisfied request for device $n$, then we use $J(t,n)$ to denote this particular request; otherwise, we set $J(t,n) = \text{NULL}$. Notice that $\mathcal{H}_t = \{(D(s), C(s), J(s)), \forall 0 \leq s \leq t\}$ completely specifies the “history” of device operations, consumer requests/cancellations and the EMS decisions until the end of time period $t$.

Generally speaking, the user’s dissatisfaction on rescheduling at time $t$ should be a function of $\mathcal{H}_t$, and we denote this function as

$$U(t)[\mathcal{H}_t] = U(t)[(D(s), C(s), J(s)), \forall 0 \leq s \leq t].$$

(1)

Obviously, directly working with $U(t)[\mathcal{H}_t]$ defined in Eqn (1) will result in a computationally intractable problem. To overcome this challenge, in the remainder of this subsection, we make some assumptions on the functional form of $U(t)[\mathcal{H}_t]$. All such assumptions are motivated by observations of the preferences and behaviors of the rational consumers. Generally speaking, we think a rational consumer’s preferences and behaviors should have the following characteristics:

1. A rational consumer prefers low electricity price to high electricity price.
2. For a consumer request sent to the EMS, a rational consumer prefers that job is completed at a time close to the target time, and the higher the request’s priority is, the “stronger” this preference is.
3. For an EMS-initiated job, a rational consumer will be “happy” if the EMS does this job at a “right” time, and will be “unhappy” if it does this job at a “wrong” time.
4. When a rational consumer decides to cancel an unsatisfied consumer request, he is usually “unhappy” if the target time has already passed.
5. A rational consumer generally discounts future costs and benefits.
6. A rational consumer’s preference over job rescheduling changes slowly over time and is reflected in his evaluations on completed/canceled jobs.
7. A rational consumer’s preference over job rescheduling on smart devices is weak compared to his preferences in other aspects of life.

We now motivate and propose a specific class of functional forms for the dissatisfaction function $U(t)[\mathcal{H}_t]$ of the rational consumers. To formalize the idea, we define the “history” for device $n$ at time $t$ as $\mathcal{H}_t^{(n)} = \{(1(n \in D(s)), 1(n \in C(s)), J(s,n)), \forall 0 \leq s \leq t\}$, where $1(\cdot)$ is the indicator function and $J(s,n)$ is defined above. First, we make the following simplifying assumption:

---

2 Note that a rational consumer’s preference over job rescheduling is weak compared to his preferences in other aspects of life. Based on this, let us provide an intuitive motivation for Assumption 1 from the following perspective: let $\tilde{u}_{DR} \in \mathbb{R}^N$ be a vector encoding the user’s dissatisfactions on job rescheduling over $N$ devices, and $\tilde{u}_{other}$ be a vector encoding the user’s dissatisfactions in other aspects of life. Assume the overall dissatisfaction (unhappiness) of the user is $f(\tilde{u}_{DR}, \tilde{u}_{other})$, where $f$ is a general non-linear function. Now we consider $f(\tilde{u}_{DR} + \Delta \tilde{u}_{DR}, \tilde{u}_{other})$, notice that the weak preference over job rescheduling implies that $\Delta \tilde{u}_{DR}$ is “small”, thus, $f(\tilde{u}_{DR} + \Delta \tilde{u}_{DR}, \tilde{u}_{other})$ can be well approximated by $f(\tilde{u}_{DR}, \tilde{u}_{other}) + \nabla f(\tilde{u}_{DR}, \tilde{u}_{other})\Delta \tilde{u}_{DR}$. Notice that for our purpose, we only care about the “change” of the user’s overall unhappiness as a function of the “changes” in his dissatisfactions on job rescheduling, thus, the user’s dissatisfaction function is $\bar{U}(\Delta \tilde{u}_{DR}) = f(\tilde{u}_{DR} + \Delta \tilde{u}_{DR}, \tilde{u}_{other}) - f(\tilde{u}_{DR}, \tilde{u}_{other}) \approx \nabla f(\tilde{u}_{DR}, \tilde{u}_{other}) \Delta \tilde{u}_{DR}$, which is a weighted sum of $\Delta \tilde{u}_{DR}$. In Assumption 1, we further simplify the dissatisfaction function by assuming that all the weights are equal to 1.
**Assumption 1**  For any \( t \geq 0 \), the dissatisfaction function \( \bar{U}^{(t)}(n) \) is additive over the devices, that is

\[
\bar{U}^{(t)}(\mathcal{H}_t) = \sum_{n=1}^{N} \bar{U}^{(t,n)}(\mathcal{H}_t^{(n)}),
\]

where \( \bar{U}^{(t,n)} \) captures the consumer’s dissatisfaction at time \( t \) for device \( n \) and \( \mathcal{H}_t^{(n)} \) is the “history” for device \( n \) by time \( t \).

Furthermore, since a rational consumer’s preference over job rescheduling is weak compared to his preferences in other aspects of life, and changes slowly over time, we assume that

- If there is no job completed/cancelled at time \( t \) on device \( n \), we assume that \( \bar{U}^{(t,n)}(\mathcal{H}_t^{(n)}) = 0 \).
- If there is a job completed/cancelled at time \( t \) on device \( n \), we assume that \( \bar{U}^{(t,n)}(\mathcal{H}_t^{(n)}) \) only depends on the job completed/canceled. That is, \( \bar{U}^{(t,n)}(\cdot) \) is time-invariant and does not depend on the previously completed/canceled jobs.

Finally, based on the assumptions above and the characteristics (2)-(4) of a rational consumer’s preferences over job rescheduling, we assume the dissatisfaction function \( \bar{U}^{(t,n)}(\mathcal{H}_t^{(n)}) \) takes the following forms:

- If device \( n \) satisfies consumer request \( J(t,n) = (n, \tau_r, \tau_g, g) \) at time \( t \), we assume

\[
\bar{U}^{(t,n)}(\mathcal{H}_t^{(n)}) = \bar{U}_r^{(n)}(t - \tau_g, g),
\]

where the subscript “\( r \)” denotes that it is the dissatisfaction incurred when a request is satisfied, and \( g \) is the priority of the request. Note \( t - \tau_g \) captures not only the distance between the current time \( t \) and the request’s target time \( \tau_g \), but also whether or not the target time has passed. For rational consumers, \( \bar{U}_r^{(n)} \) should be small when \( t - \tau_g \) is close to 0 and increases as \( t - \tau_g \) deviates from 0; furthermore, the higher the priority \( g \), the higher this increase rate will be.

- Similarly, if the request \( J(t,n) = (n, \tau_r, \tau_g, g) \) is cancelled by the consumer at time \( t \), we assume

\[
\bar{U}^{(t,n)}(\mathcal{H}_t^{(n)}) = \bar{U}_c^{(n)}(t - \tau_g, g),
\]

where the subscript “\( c \)” denotes that it is the dissatisfaction incurred when a request is cancelled. In practice, \( \bar{U}_c^{(n)} \) should be very small if \( t < \tau_g \), and it will increase with \( t \) when \( t \geq \tau_g \); furthermore, the higher the priority \( g \), the higher this increase rate will be.

- If EMS initiates a job on device \( n \) at time \( t \), we assume that

\[
\bar{U}^{(t,n)}(\mathcal{H}_t^{(n)}) = \bar{U}_e^{(n)}(t - \tau_p),
\]

where the subscript “\( e \)” denotes that it is the dissatisfaction incurred when an EMS-initiated job is done, and \( \tau_p - 1 \) is the time when the previous job on device \( n \) (either user-initiated or EMS-initiated) is completed or cancelled. In practice, \( \bar{U}_e^{(n)} \) should decrease as \( t - \tau_p \) increases; the large \( \bar{U}_e^{(n)} \) for small \( t - \tau_p \) prevents too frequent EMS-initiated jobs.
• Otherwise, we assume that
\[ \tilde{U}^{(t,n)}(\mathcal{H}_t^{(n)}) = 0. \] (6)

Note that the dissatisfaction function class described in Equations (3-6) is still quite general, and it is very challenging to proceed to derive the specific functional forms of \( \tilde{U}_r^{(n)}, \tilde{U}_c^{(n)} \) and \( \tilde{U}_e^{(n)} \). As we will see in Section 4, there is no actual need to estimate \( \tilde{U}_r^{(n)}, \tilde{U}_c^{(n)} \) and \( \tilde{U}_e^{(n)} \). In practice, the EMS learns \( \tilde{U}_r^{(n)}, \tilde{U}_c^{(n)} \) and \( \tilde{U}_e^{(n)} \) based on the consumer’s evaluations on completed/canceled jobs.

3.2 Instantaneous Cost Function

We assume the instantaneous cost function of the consumer at time \( t \) has the following form:
\[ P_t \sum_{n \in \mathcal{D}(t)} C_n + \gamma \tilde{U}^{(t)}(\mathcal{H}_t), \] (7)
where \( P_t \) is the electricity price at time \( t \) and \( \tilde{U} \) is a dissatisfaction function capturing the consumer’s dissatisfaction on job rescheduling at time \( t \). Specifically, notice that \( \sum_{n \in \mathcal{D}(t)} C_n \) is the total electricity energy consumed at time \( t \), and hence \( P_t \sum_{n \in \mathcal{D}(t)} C_n \) is the electricity bill the consumer pays at time \( t \). We assume the consumer’s dissatisfaction on rescheduling at time \( t \) depends on the “rescheduling history” \( \mathcal{H}_t \) and \( \gamma > 0 \) represents the tradeoff between the electricity bill paid and the consumer’s dissatisfaction on rescheduling.

Note that from the EMS’s perspective, both the electricity price and the consumer behavior are exogenous and stochastic; thus, in this paper, we assume that EMS aims to minimize the expected infinite-horizon discounted cost:
\[ E \left\{ \sum_{t=0}^{\infty} \alpha^t \left[ P_t \sum_{n \in \mathcal{D}(t)} C_n + \gamma \tilde{U}^{(t)}(\mathcal{H}_t) \right] \right\}, \]
where \( 0 < \alpha < 1 \) is the discrete-time discount and captures the assumption that a rational consumer generally discounts future costs and benefits. Notice that in this problem formulation, the state at time \( t \) is \( (P_t, \mathcal{H}_{t-1}, \mathcal{J}(t)) \), and the action is \( \mathcal{D}(t) \) (i.e. the EMS chooses which devices to power on).

Under Assumption 1, the dissatisfaction function is decomposable and the infinite-horizon discounted cost function can be written as
\[ \sum_{n=1}^{N} E \left\{ \sum_{t=0}^{\infty} \alpha^t \left[ P_t C_n 1(n \in \mathcal{D}(t)) + \gamma \tilde{U}^{(t,n)}(\mathcal{H}_t^{(n)}) \right] \right\}. \] (8)

3.3 Device Based MDP Model

In this subsection, we first propose a dynamic model for the energy price \( P_t \). Then, we motivate an assumption on user behavior (Assumption 3), and show that under the proposed assumptions, the EMS’s decision-making (job scheduling) problem is decomposed over devices. Finally, we formulate the job scheduling problem on each device as a Markov Decision Process (MDP), which we henceforth refer to as the device based MDP model.

Let us start by proposing a dynamic model for the energy price \( P_t \). As discussed in Section 2, \( P_t \) is exogenous in the sense that the EMS’s actions will not influence it. Furthermore, recent literature (e.g. [ONeill et al., 2010]) shows that \( P_t \) can be modeled as a Markov process. Thus, throughout this paper, we make the following assumption:
**Assumption 2**  *Energy price* $P_t$ *follows an exogenous finite-state ergodic Markov chain.*

We use $\mathcal{P}$ to denote the set of states of this Markov chain, and use $|\mathcal{P}|$ to denote the number of states.

In this paper, we assume that the user is rational and the user’s dissatisfaction function has the functional form specified by Eqn(3-6). Thus, to complete the description of the user behavior, we only need to model when the user sends requests/cancellations to the EMS. Similarly as $P_t$, we model the consumer requests/cancellations as stochastic processes and make the following assumption on them throughout this paper:

**Assumption 3** (A) The consumer request/cancellation processes are statistically independent of the energy price process $P_t$; and (B) consumer requests/cancellations to different devices are statistically independent.

Assumption 3 is a key assumption in this paper. Thus, we briefly motivate it before proceeding: first, we claim that Assumption 3(A) is reasonable since

- On one hand, as we have discussed in Section 2, in the residential and small building sector, the market power of an individual consumer is so small that the impact of their behavior on $P_t$ is negligible.
- On the other hand, as we have discussed in Section 1, due to “decision fatigue”, most rational consumers are not incentivized to reference current and expected future energy prices before sending requests/cancellations to the EMS. Thus, for such consumers, their requests/cancellations are independent of current and expected future energy prices.

Consequently, it is reasonable to assume Assumption 3(A).

We now briefly motivate Assumption 3(B). One observation about devices is that they can be classified into *categories* such that for most consumers, their requests to devices in different categories are weakly dependent. For instance, in a residential household, we can classify the air-conditioner, electric vehicle, laundry machine and dryer into three categories: category 1 includes the air-conditioner; category 2 includes the laundry machine and dryer; and category 3 includes the electric vehicle. Under this classification, it is observed that for most residential consumers, the statistical dependence between their requests to devices in different categories are weak. Thus, it is reasonable to assume that consumer requests/cancellations to different device categories are statistically independent. To simplify, we assume each category has only one device and hence propose Assumption 3(B). Of course, a category might have multiple devices (e.g. laundry machine and dryer), and usually the consumer requests to devices in the same category are statistically dependent. We discuss how to extend the results of this paper to that scenario in Section 6.

Recall that the EMS’s objective is to find an optimal scheduling policy to minimize (8), which decomposes over devices under Assumption 1. Assumption 3 states that the consumer requests/cancellations to different devices are statistically independent, thus, we have the following proposition:

**Proposition 1** Under Assumptions 1 and 3, the EMS’s decision-making problem is decomposed over devices.

In the remainder of this paper, we focus on deriving the optimal scheduling policy for a single device and drop the subscript $n$ to simplify the exposition. For example, we will use $W$ instead of $W_n$ to denote the time window and represent a request as $J = (\tau_r, \tau_g, g)$. We also use the term “smart
device” and “EMS” interchangeably henceforth, since due to the decomposition of the problem, one can think each smart device has its own EMS.

We further assume that the timeline for a smart device can be divided into “episodes”. Specifically, we assume that whenever the smart device completes a job (either initiated by the user or by the EMS) or the current unsatisfied request is canceled by the user, the current episode terminates. In the next time step, the smart device “regenerates” its state according to a fixed distribution $\pi_0$ and a new episode starts. The notion of episode is illustrated in Figure 3.

![Figure 3: Illustration of the notion of episode](image)

We now propose an MDP model for a single smart device (the device based MDP model). The state of the MDP at time $t$ is $x_t = [P_t, s_t, g_t]^T \in S$, where $P_t$ is the exogenous electricity price at time $t$, $s_t$ is the elapsed time at time $t$, $g_t$ is the priority of request at time $t$ and $S$ is the state space. Specifically, we define

$$
s_t = \begin{cases} 
  t - \tau_p & \text{if no request received in the current episode} \\
  t - \tau_g & \text{otherwise}
\end{cases}
$$

where $\tau_p$ is the start time of the current episode and $\tau_g$ is the target time of the received request. Furthermore, we use $g_t = 0$ to denote that no request has yet been received in the current episode; once a request is received in the current episode, we assume its priority $g_t \in \{1, 2, \cdots, g_{\text{max}}\}$.

Since the electricity price is exogenous, we can partition the state $x_t = [P_t, s_t, g_t]^T$ as the “price portion” $P_t$ and “device portion” $[s_t, g_t]^T$. The “device portion” of the MDP state transition model is summarized in Figure 4; notice that there are $(2W + 1)g_{\text{max}} + W + 1$ “device portion” states. Recall that we use $\mathcal{P}$ to denote the set of states of the price Markov chain, so the price Markov chain has $|\mathcal{P}|$ states, and the cardinality of the state space for the device based MDP is $|S| = |\mathcal{P}| \left[(2W + 1)g_{\text{max}} + W + 1\right]$, which is affine in $|\mathcal{P}|$, $W$, $\hat{W}$ and $g_{\text{max}}$.

Note that the action space at each state $x \in S$ is $\mathcal{A} = \{\text{off}, \text{on}\}$, where action “on” means the smart device completes a job at the current time and “off” means the smart device does nothing at the current time. We use $\Phi(x_t, a_t, x_{t+1})$ to denote the instantaneous cost at state-action-state triple $(x_t, a_t, x_{t+1})$.

Based on the above discussion, the device based MDP model is detailed below:

- $P_t$ follows an exogenous Markov Chain with $|\mathcal{P}|$ states. The transition probability from $P_t$ to $P_{t+1}$ is denoted as $\Pr(P_{t+1}|P_t)$.
- If the smart device has not received a consumer request in the current episode, recall that we set $g_t = 0$ and elapsed time $s_t = t - \tau_p$. Notice that:
  - 1. If action “off” is selected, the current cost is $\Phi(x_t, a_t, x_{t+1}) = 0$. Then the smart device receives a consumer request $(t + 1, \tau_g, g)$ at the next time step (time $t + 1$) with

---

3More specifically, it completes a user-initiated job if a consumer request has been received in the current episode, and completes an EMS-initiated job otherwise.
probability \(p_{s_t, t+1 - \tau_g, g}\), where \(t + 1 \leq \tau_g \leq t + 1 + W\) and \(g \in \{1, 2, \ldots, g_{\text{max}}\}\). In other words, the “device portion” state transits to \([s_{t+1} = t + 1 - \tau_g, g_{t+1} = g]^T\) with probability \(p_{s_t, t+1 - \tau_g, g}\), for any \(t+1 \leq \tau_g \leq t+1+W\) and any \(g \in \{1, 2, \ldots, g_{\text{max}}\}\) (notice that the target time and priority together specify the type of the received request). On the other hand, with probability \(1 - \sum_{\tau_g=t+1}^{t+1+W} \sum_{g=1}^{g_{\text{max}}} p_{s_t, t+1 - \tau_g, g}\), the smart device does not receive the consumer request at time \(t+1\) and the “device portion” state transits to \([s_{t+1} = s_t, g_{t+1} = 0]^T\).

2. On the other hand, if action “on” is selected, then the smart device completes an EMS-initiated job at time \(t\). Thus, the current cost is \(\Phi(x_t, a_t, x_{t+1}) = P_t C + \gamma \tilde{U}_e(s_t)\), where \(\tilde{U}_e(s_t)\) is defined in (5) and \(C\) is the constant energy consumed by a standardized job. Then the current episode terminates and the smart device regenerates the “device portion” state based on distribution \(\pi_0\).

Note that the transition probability \(p_{s_t, t+1 - \tau_g, g}\) and the dissatisfaction \(\tilde{U}_e(s_t)\) depend on \(s_t\). To ensure the state space is finite, we assume \(p_{s_t, t+1 - \tau_g, g} = p_{W_t, t+1 - \tau_g, g}\) and \(\tilde{U}_e(s_t) = \tilde{U}_e(W)\) for any \(s_t \geq W\).

- If the smart device has already received a consumer request in the current episode but has

\[\text{That is, when } s_t \geq W \text{ and action “off” is selected, with probability } 1 - \sum_{\tau_g=t+1}^{t+1+W} \sum_{g=1}^{g_{\text{max}}} p_{W_t, t+1 - \tau_g, g}, \text{ the smart device will stay at the same state.}\]
not satisfied this request, recall we set $s_t = t - \tau_g$, where $\tau_g$ is the target time of this request. Notice that:

1. If action “off” is selected, then with probability $1 - \tilde{p}_{st,gt}$, the user will cancel this request at the end of time period $t$. In this case, the current episode terminates and the smart device regenerates the “device portion” state based on distribution $\pi_0$ at time $t + 1$. The cost associated with this transition is $\Phi(x_t, a_t, x_{t+1}) = \gamma \tilde{U}_c(s_t, g_t)$, where $\tilde{U}_c$ is defined in (4). On the other hand, with probability $\tilde{p}_{st,gt}$, the “device portion” state transits to $[s_{t+1} = s_t + 1, g_{t+1} = g_t]^T$ at time $t + 1$ and the cost associated with this transition is $\Phi(x_t, a_t, x_{t+1}) = 0$. Notice that we assume $\tilde{p}_{st,gt} = 0$ if $s_t = W$.

2. If action “on” is selected, the current cost is $\Phi(x_t, a_t, x_{t+1}) = P_tC + \gamma \tilde{U}_r(s_t, g_t)$, where $\tilde{U}_r(s_t, g_t)$ is defined in (3) and $C$ is the constant energy consumed by a standardized job. Then the current episode terminates and the smart device regenerates the “device portion” state based on distribution $\pi_0$ at time $t + 1$.

Note that if the transition model of the device based MDP and the dissatisfaction function of the consumer are known, the device based MDP can be solved by dynamic programming (DP). Following ideas in classical DP, it is straightforward to derive the Bellman equation (see Appendix A) from which we can compute the optimal Q-function $Q^*$. Specifically, many DP algorithms, such as value iteration and policy iteration, can be used to compute $Q^*$ (see [Bertsekas, 2005]). We observe that for many DP algorithms, computing $Q^*$ is tractable since the cardinalities of the state space $S$ and action space $A$ in a device based MDP are usually small. Once $Q^*$ is available, one optimal policy $\mu^*$ is $\mu^*(x_t) \in \arg\min_{a \in A} Q^*(x_t, a)$.

4 Reinforcement Learning and Performance Metrics

4.1 Reinforcement Learning Formulation

However, as we discussed in Section 1, in most practical cases, the transition probabilities and the dissatisfaction function of the user are initially unknown. Thus, the EMS must learn how to make optimal decisions for the user based on the incrementally gathered data from the user and the real-time energy price. Reinforcement learning (RL) is a collection of techniques for a decision-maker to learn how to make optimal decisions while interacting with an unknown “environment” [Sutton and Barto, 1998]. RL tries to strike a balance between learning (exploration) and optimization (exploitation), and has been extensively used in many other fields, such as artificial intelligence and robotics [Coates et al., 2010]. As has been discussed in [O’Neill et al., 2010], it is natural to use RL to solve EMS’s learning problem since EMS needs to learn to make decisions while interacting with the user and the real-time energy price. As is shown in Figure 5, under the RL formulation of the optimal DR problem, the agent is the EMS, and the environment includes both the energy price and consumer behavior. We refer to this RL problem as the device based RL problem.

As in the RL literature, we assume the EMS (decision-maker) initially knows the state space $S$ and the action space $A = \{\text{off, on}\}$. Furthermore, we also assume it knows the discrete-time discount $\alpha$, and the price-rescheduling trade-off parameter $\gamma$. Note that the second assumption is reasonable since before the user starts to use the smart device, they can manually set up an $\alpha$ reflecting their discount on future costs, as well as a $\gamma$ indicating their tradeoff between the electricity bill and rescheduling. With experience, the user can also tune these parameters.

However, as is discussed above, in theory the EMS (the decision-maker) needs to learn the transition model and the dissatisfaction function of the user based on its experience. Obviously,
the transition model can be learned by directly observing the user behavior and the real-time energy price. We now discuss how the EMS learns the user’s dissatisfaction function. Note that under our proposed device based MDP model, the user’s dissatisfaction function is non-zero only when a job is completed or canceled. Furthermore, notice that as has been discussed in Section 2, once a job is completed/canceled, the EMS will receive an evaluation from the user. In the remainder of this paper, we make the following assumption:

Assumption 4: The user’s dissatisfaction on the rescheduling of a completed/canceled job is equal to their evaluation on this job.\(^5\)

We now briefly motivate Assumption 4: for a rational user, their evaluation on a completed/canceled job mainly reflects their dissatisfaction on this job, and this dissatisfaction is either due to the (high) energy price or the (undesirable) rescheduling of the job. As we have discussed in Section 1, due to “decision fatigue”, most rational users will not reference the current energy price or any expected future energy prices before they send their evaluations to the EMS. Thus, their evaluations will mainly reflect their dissatisfactions on job rescheduling. Thus, it is reasonable to assume Assumption 4.

In summary, in our proposed device based RL problem, the EMS initially knows the state space \(S\), the action space \(A\), the discount \(\alpha\), the trade-off parameter \(\gamma\) and the per-job energy consumption \(C\); but it does not know the state transition model or the user’s dissatisfaction function. It observes the state transitions and the user’s evaluations (which are equal to the user’s dissatisfactions on rescheduling under Assumption 4) as it interacts with the user and the real-time energy price. At each time step, it aims to make good decisions based on its initial information, past observations and current state.

4.2 Baseline and Demand Response Potential

In order to justify a RL algorithm achieves satisfactory performance, we need to compare its experimental performance with respect to a reasonable baseline. In this subsection, we propose a reasonable baseline and define the notion of the demand response (DR) potential.

Let’s start by defining some useful notation: we use \(\mu : S \times A \rightarrow [0, 1]\) to denote a (randomized and stationary) policy of a device based MDP. Specifically, under (randomized) policy \(\mu\), at any

\(^5\)Assumption 4 is an idealized assumption on the relationship between evaluation and dissatisfaction. In practice, it might be more reasonable to assume that the user’s evaluation on a job is equal to their dissatisfaction on the rescheduling of this job plus a (zero-mean, finite-variance, statistically independent) “behavioral noise”. However, the “behavioral noise” does not fundamentally change the assumption and we make Assumption 4 to simplify exposition.
state $x = [P, s, g]^T \in \mathcal{S}$, action $a \in \mathcal{A}$ is chosen with probability $\mu(x, a)$. As is classical in DP and RL, we use $Q_\mu$ to denote the Q-function of the device based MDP under policy $\mu$ (see [Sutton and Barto, 1998]). Note that under Assumption 2, the energy price $P_t$ follows an exogenous ergodic Markov chain; thus, we use $\pi_P$ to denote the unique stationary distribution of this price Markov chain. Moreover, recall that once an episode terminates, the “device portion” state is regenerated according to distribution $\pi_0$. Recall that state $x = [P, s, g]^T$, thus, we use $x \sim \pi_P \times \pi_0$ to denote $P \sim \pi_P$, $[s, g]^T \sim \pi_0$, and $P$ and $[s, g]^T$ are statistically independent. We define the performance of a policy $\mu$ as follows:

$$V_\mu = \mathbb{E}_{x \sim \pi_P \times \pi_0} \left\{ \mathbb{E}_{a \sim \mu(x, \cdot)} [Q_\mu(x, a) | x] \right\}. \quad (9)$$

Since this paper focuses on how demand response (DR) can potentially reduce a user’s (expected infinite-horizon discounted) cost, we choose the baseline policy as the one without job rescheduling and denote it as $\mu_{\text{base}}$. Specifically, under $\mu_{\text{base}}$, the EMS never self-initiates a job, and all the jobs initiated by the user are scheduled to be completed at their target times. We are interested in how much a proposed RL algorithm can reduce the user’s cost with respect to $V_{\mu_{\text{base}}}$.

Recall that we use $\mu^*$ to denote the optimal policy. We define the demand response (DR) potential as

$$\text{DRP} = V_{\mu_{\text{base}}} - V_{\mu^*}. \quad (10)$$

By definition, DR potential is the maximum expected cost reduction that can be achieved by DR. Obviously, in the case when the transition model and the dissatisfaction function are known, $\mu^*$ can be derived beforehand and hence DRP is achieved. However, in the practical cases when the EMS needs to learn $\mu^*$ through some RL algorithm, DRP is generally not achievable. As we will see, performance loss with respect to DRP is one performance metric for RL algorithms.

We define the relative DR potential (RDRP) as

$$\text{RDRP} = \frac{\text{DRP}}{V_{\mu_{\text{base}}}} = \frac{(V_{\mu_{\text{base}}} - V_{\mu^*})}{V_{\mu_{\text{base}}}}. \quad (11)$$

Of course, DRP and RDRP depend on the problem instance. We are particularly interested in how DRP and RDRP vary with $\gamma$, since $\gamma$ specifies the consumer’s tradeoff between job rescheduling and the electricity bill paid. To simplify the expositions of the result, we make the following assumption in the remainder of the paper:

**Assumption 5** (A) Energy price is always strictly positive; (B) the user’s dissatisfaction function satisfies Eqn(3-6) and is always non-negative; (C) if a user’s request is satisfied at its target time, the user’s dissatisfaction is 0; and (D) if the user cancels their request before the target time, their dissatisfaction is 0.

Note that Assumption 5(C) and 5(D) are reasonable since for a rational user, it is proper to assume their dissatisfaction is minimal if their request is satisfied at the target time, or canceled before the target time. Then Assumption 5(B) can be achieved by shifting the dissatisfaction function by a constant. Assumption 5(A) follows from the daily observation. Under Assumption 5, we have the following result:

**Theorem 1** Under Assumption 5, we have that:

1. $V_{\mu_{\text{base}}}$ does not depend on $\gamma$.

2. There exists a $\gamma^* > 0$ s.t. $\mu_{\text{base}}$ is an optimal policy when $\gamma > \gamma^*$. Thus, as $\gamma \to \infty$, DRP $\to 0$. 

15
3. DRP and RDRP are non-increasing functions of $\gamma$.

4. If $\gamma = 0$, then $RDRP = 1$.

Please refer to Appendix B for the proof of Theorem 1.

4.3 Performance Metrics

In this subsection, we propose several performance metrics for a RL algorithm applied to our proposed device based RL problem. We start by defining some notation: for any $t = 0, 1, \cdots$, use $\tilde{\mu}_t$ to denote the policy under which the RL algorithm chooses the action $a_t$ at the beginning of time period $t$. We define the performance of the RL algorithm as

$$
\tilde{V} = \mathbb{E} \left[ \sum_{t=0}^{\infty} \alpha^t \Phi (x_t, \tilde{\mu}_t (x_t), x_{t+1}) \right],
$$

(12)

note that the expectation is not only taken with respect to the initial state (similarly as Eqn(9), we assume $x_0 = [P_0, s_0, g_0]^T \sim \pi_P \times \pi_0$ in Eqn(12)), but also with respect to the subsequent stochastic transitions, noisy evaluations and possible randomizations in the RL algorithm. Note that by definition, we have $V_{\mu^*} \leq \tilde{V}$.

The first performance metric of an RL algorithm is its relative improvement (RI) with respect to the baseline, which is defined as

$$
RI = (V_{\mu_{base}} - \tilde{V})/V_{\mu_{base}}.
$$

(13)

RI captures the normalized expected cost reduction the user will benefit from an RL algorithm. Note that since $V_{\mu^*} \leq \tilde{V}$, we have $RI \leq RDRP$. That is, the relative DR potential is an upper bound on the relative improvement.

The second performance metric of an RL algorithm is its relative realized DR potential (RRDRP), which is defined as

$$
RRDRP = RI/RDRP = (V_{\mu_{base}} - \tilde{V})/(V_{\mu_{base}} - V_{\mu^*}),
$$

(14)

if $RDRP > 0$. Intuitively, RRDRP captures how much DR potential is realized by an RL algorithm. Since $V_{\mu^*} \leq \tilde{V}$, we have $RRDRP \leq 1$.

We briefly comment on $RI$ and $RRDRP$ before proceeding. First, notice that both $RI$ and $RRDRP$ can be negative, since there is no guarantee that $\tilde{V} \leq V_{\mu_{base}}$. Specifically, under Assumption 5, it is likely that $\tilde{V} > V_{\mu_{base}}$ when $\gamma$ is sufficiently large. This is because $V_{\mu_{base}} \downarrow V_{\mu^*}$ as $\gamma \to \infty$, and whatever $\gamma$ is, an RL algorithm is likely to incur a non-trivial performance loss before it learns an optimal policy.

Second, we observe that RRDRP also serves as an indicator on whether or not it is worthy to explicitly model the user behavior/dissatisfaction and the energy price. Specifically, if $DRP$ is large but $RRDRP$ is small for many widely used RL algorithms, then it might be worthy to explicitly model the user behavior/dissatisfaction and the energy price, as long as the cost of such modeling is smaller than $\tilde{V} - V_{\mu^*}$.

We also note that it is important for our proposed RL algorithm to achieve satisfactory performance quickly, since otherwise the consumer will not be incentivized to use it. The third performance metric focuses on this aspect. Specifically, $\forall j = 0, 1, \cdots$, we use $t_j$ to denote the end time
of episode \( j \). We define
\[
V_{\mu_{\text{base}}}(j) = E \left[ \sum_{t=0}^{t_j} \alpha^t \Phi(x_t, \mu_{\text{base}}(x_t), x_{t+1}) \right].
\]
Thus, \( V_{\mu_{\text{base}}}(j) \) is the user’s expected total discounted cost in the first \( j + 1 \) episodes (from episode 0 to episode \( j \)) under the baseline policy. Similarly, we define
\[
\tilde{V}(j) = E \left[ \sum_{t=0}^{t_j} \alpha^t \Phi(x_t, \tilde{\mu}_t(x_t), x_{t+1}) \right],
\]
where \( \tilde{\mu}_t \) is the policy under which the RL algorithm chooses the action \( a_t \) at the beginning of time period \( t \). Similarly, \( \tilde{V}(j) \) is the user’s expected total discounted cost in the first \( j + 1 \) episodes under the RL algorithm.

We define the third performance metric, the baseline crossing time, as
\[
\text{BCT} = \min \left\{ j^* \geq 0 : \tilde{V}(j) \leq V_{\mu_{\text{base}}}(j), \forall j \geq j^* \right\}. \tag{15}
\]
Note that BCT is measured in the number of episodes instead of the number of time periods. This is because measuring time in episodes is more meaningful in our problem: each episode corresponds to one usage of the device, and hence the number of episodes corresponds to how many times the consumer has used the device. Thus, measuring time in episodes avoids the problem that the consumer might sometimes use the device more frequently, while at other times use it less frequently.

5 Simulation

In this section, we first briefly review the classical Q-learning algorithm in Subsection 5.1. Then we discuss a representative simulation example in Subsection 5.2. Finally, the simulation results are demonstrated in Subsection 5.3.

5.1 The Q-Learning Algorithm

As we have discussed in Section 4, many RL algorithms can be applied to our proposed device based RL problem. In this subsection, we implement one of the most popular and classical RL algorithms, known as Q-learning [Watkins, 1989]. Q-learning is an off-policy learning algorithm; that is, it allows the learning agent to follow an exploratory policy while learning about an optimal policy. Another desirable features of Q-learning is that it is online, incremental and is easy to implement on real-time data.

Q-learning works based on temporal-difference learning [Sutton and Barto, 1998]. At each time-step \( t \) the learning algorithm receives an input data in the form of \((x_t, a_t, \Phi_t, x_{t+1})\), where \( \Phi_t \equiv \Phi(x_t, a_t, x_{t+1}) \) is the observed instantaneous cost after taking action \( a_t \) from state \( x_t \) and arriving at state \( x_{t+1} \). Then the Q-learning algorithm updates the action-value function \( Q_t(x_t, a_t) \) according to
\[
Q_{t+1}(x_t, a_t) := Q_t(x_t, a_t) + \beta_t \left[ \Phi_t + \alpha \min_{a' \in A} Q_t(x_{t+1}, a') - Q_t(x_t, a_t) \right],
\]
\(^{6}\)Note that \( t_j \)'s are random variables (more specifically, stopping times). Furthermore, the distribution of \( t_j \)'s depends on the policy.
and \( Q_{t+1}(x, a) := Q_t(x, a) \) if \((x, a) \neq (x_t, a_t)\). Note \( Q_t \in \mathbb{R}^{|S| \times |A|} \) is the state-action value function (Q-function) estimate in time period \( t \), and \( \beta_t > 0 \) denotes step-size in time period \( t \). If the step-size sequence satisfies \( \sum_{t=0}^{+\infty} \beta_t = +\infty \) and \( \sum_{t=0}^{+\infty} \beta_t^2 < +\infty \), then Q-learning is guaranteed to converge to the optimal solution if all states are visited infinitely often (see [Sutton and Barto, 1998]).

To complete the description of a Q-learning algorithm, we also need to specify a behavioral policy \( \mu_b \) under which the algorithm chooses actions. The choice of \( \mu_b \) will affect data and thus would help the algorithm to learn an optimal policy faster. One main ingredient of \( \mu_b \) is that it has to be exploratory policy during the learning process. One of standard, but crude, suggestions for how to select \( \mu_b \) is as follows: for a small \( 0 < \epsilon < 1 \), with probability \( \epsilon \) (henceforth referred to as the “exploration probability”), the algorithm chooses action \( a_t \) from state \( x_t \) according to a randomized policy, and with probability \( 1 - \epsilon \), the algorithm chooses \( a_t \in \text{argmin}_{a \in A} Q_t(x_t, a) \), where \( Q_t \) refers to the state-action value function estimate at time \( t \). Here, we consider a randomized policy in the form of Boltzmann exploration

\[
\pi(a_t|x_t) = \frac{\exp \left[-Q_t(x_t, a_t)/\eta\right]}{\sum_{a' \in A} \exp \left[-Q_t(x_t, a')/\eta\right]},
\]

where \( \eta > 0 \) is a tuning parameter and is referred to as the “temperature” of the Boltzmann exploration. The Q-learning algorithm that we implement in this paper is illustrated in Algorithm 1.

**Algorithm 1 The Q-Learning Algorithm**

1: \textbf{Initialize} \( Q_0 \) arbitrarily
2: \textbf{Repeat} for each episode \( j \):
3: \hspace{1em} Choose a small constant step-size \( \beta_j > 0 \) for each episode
4: \hspace{1em} for each time period \( t \) in episode \( j \) \textbf{do}
5: \hspace{2em} Take action \( a_t \) at state \( x_t \) according to the behavioral policy \( \mu_b \)
6: \hspace{2em} Observe the instantaneous cost \( \Phi_t \) and new state \( x_{t+1} \)
7: \hspace{2em} Compute the TD error \( \delta_t := \Phi_t + \alpha \min_{a' \in A} Q_t(x_{t+1}, a') - Q_t(x_t, a_t) \)
8: \hspace{2em} Update \( Q_{t+1}(x, a) := \begin{cases} Q_t(x, a) + \beta_j \delta_t & \text{if } (x, a) = (x_t, a_t) \\ Q_t(x, a) & \text{otherwise} \end{cases} \)
9: \hspace{1em} \textbf{end for}

**5.2 Simulation Setup**

In this subsection, we propose a representative example to which we apply the Q-learning algorithm detailed in Algorithm 1. Specifically, in this example, we assume that the exogenous price Markov chain has state space \( \mathcal{P} = \{10, 12, 15, 20\} \), and the consumer requests have two different priorities, “high” and “normal”. We set the time windows \( W = 4 \) and \( \bar{W} = 5 \), the discrete-time discount \( \alpha = 0.9995 \) and the per job energy consumption \( C = 1 \). Thus, there are \(|S| = 96\) states in this example. The dissatisfaction functions \( \tilde{U}_r \), \( \tilde{U}_c \) and \( \tilde{U}_e \) are illustrated in Figure 6(a), 6(b) and 6(c). Notice that these dissatisfaction functions satisfy the characteristics of the rational consumer preferences discussed in Subsection 3.1, as well as Assumption 5.
As to the transition model, we assume that if the smart device has not received a consumer request in the current episode, then under action “off”, it will receive a consumer request in the next time step with probability \( p_s \). Notice that \( p_s \) is chosen to be an increasing function of the elapsed time \( s_t \) (see Figure 6(d)). We further notice that there are \((W + 1)g_{\text{max}} = 10\) types of consumer requests (with different target times and priorities), for simplicity, we assume these \(10\) types of requests are equally likely. Furthermore, if the smart device has received a consumer request in the current episode, then under action “off”, the unsatisfied request will be canceled with probability \( \hat{p}_s \). In this example, we assume the “cancellation probability” \( \hat{p} \) only depends on the elapsed time \( s_t \) and is independent of the priority \( g_t \). We choose \( \hat{p}_s \) as an increasing function of \( s_t \) (see Figure 6(e)). Finally, we assume that when the smart device regenerates its state, with probability 1, the regenerated “device portion” state is \([s_{t+1} = 0, g_{t+1} = 0]^T\).

We plot the RDRP of this example as a function of the trade-off parameter \( \gamma \) (see Figure 7). Since this example satisfies Assumption 5, thus, as Theorem 1 indicates, the RDRP is a non-increasing function of \( \gamma \). Furthermore, when \( \gamma = 0 \), \( \text{RDRP} = 1 \), and \( \text{RDRP} \to 0 \) as \( \gamma \to \infty \).

### 5.3 Performance

In this subsection, we present the simulation results of the Q-learning algorithm on the representative example detailed in Subsection 5.2. We start by describing how we implement the Q-learning algorithm. We choose the “exploration probability” \( \epsilon = 0.05 \) and the “temperature” of the softmin policy \( \eta = 0.1 \). For episode \( j \), we choose the step-size \( \beta_j = \frac{10}{20(j+1)} \). We initialize the Q-learning algorithm by setting \( Q_0 = 0 \).

For each trade-off parameter \( \gamma = 0, 0.1, 0.2, \ldots \), we run the Q-learning algorithm for \( \left\lceil \frac{2}{1-\alpha} \right\rceil = 4,000 \) episodes, and repeat the simulation for 200 times. Then we approximate \( \tilde{V} \) and \( \tilde{V}(j) \),
\( \forall j = 0, 1, \cdots \) by their corresponding sample means, i.e., by averaging the simulation results in these 200 simulations. Note that both \( V_{\mu_{\text{base}}} \) and \( V_{\mu_{\text{base}}(j)} \), \( \forall j = 0, 1, \cdots \), can be analytically derived based on the Bellman equation under policy \( \mu_{\text{base}} \). Thus, RI, RRDP and BCT can be (approximately) computed based on \( V_{\mu_{\text{base}}} \), \( V_{\mu_{\text{base}}(j)} \) and the sample means of \( \tilde{V} \) and \( \tilde{V}(j) \). The simulation results are summarized in Figure 8-10, where we plot the RI, RRDP and BCT as functions of the trade-off parameter \( \gamma \). Note that we only plot the simulation results for \( 0 \leq \gamma \leq 4.3 \), since for \( \gamma \geq 4.4 \), we have \( \tilde{V} > V_{\mu_{\text{base}}} \). Consequently, for \( \gamma \geq 4.4 \), RI and RRDP are negative, and BCT = \( \infty \).

We conclude this section by briefly discussing about the simulation results. Notice that Figures 8-10 show that in this representative example, RI and RRDP are decreasing functions of \( \gamma \), while BCT is an increasing function of \( \gamma \). This implies that with all the other parameters fixed, the more the user prefers “no rescheduling” to “low energy price”, the harder for the Q-learning algorithm to (1) achieve a significant improvement compared with the baseline, (2) realize a large fraction of the DR potential, and (3) quickly outperform the baseline policy.

We also observe that BCT = 0 for \( \gamma \leq 2.1 \). This implies that for such \( \gamma \)'s, by initializing \( Q_0 = 0 \), the Q-learning algorithm outperforms the baseline policy from episode 0. Furthermore, as \( \gamma \uparrow 4.4 \), \( \tilde{V} \uparrow V_{\mu_{\text{base}}} \) and hence BCT \( \rightarrow \infty \).
Figure 9: Relative realized DR potential (RRDP) as a Function of $\gamma$

Figure 10: BCT as a Function of $\gamma$
6 Conclusion

We have motivated and proposed a novel EMS formulation for the DR problem in the residential and small commercial building sectors, which we refer to as the device based RL problem. Specifically, we have shown that under appropriate assumptions, our proposed EMS formulation does not require a pre-specified disutility function modeling the consumer’s dissatisfaction on job rescheduling, and has a computational complexity that grows linearly with the number of devices. Our new EMS formulation also enables the EMS to self-initiate jobs and allows the users to initiate more flexible requests.

We have also motivated and proposed several performance metrics for RL algorithms applied to the device based RL problem, and demonstrated the simulation results when the classical Q-learning algorithm is applied to a representative example. Simulation results suggest that for a broad range of trade-off parameter $\gamma$, the Q-learning algorithm outperforms the baseline policy without any job rescheduling.

Finally, it is worth pointing out that many assumptions of the device based MDP model discussed in this paper are proposed to simplify the exposition, and we can easily relax some of these assumptions to obtain a more general device based MDP model. In Appendix C, we discuss two relaxations that are of particular interest: stacking request and dependent devices.

Acknowledgment

We acknowledge Prof. Benjamin Van Roy for his insightful comments on this paper.

References

G. Barbose, C. Goldman, and B. Neenan. A survey of utility experience with real time pricing. Lawrence Berkeley National Laboratory: Lawrence Berkeley National Laboratory, 2004. URL http://www.escholarship.org/uc/item/8685983c.

D. Bertsekas. Dynamic Programming and Optimal Control. Athena Scientific, Massachusetts, 2005.

S. Borenstein, M. Jaske, and A. Rosenfeld. Dynamic pricing, advanced metering, and demand response in electricity markets. UC Berkeley: Center for the Study of Energy Markets, October 2002. URL http://www.escholarship.org/uc/item/11w8d6m4.

S. Braithwait and K. Eakin. The role of demand response in electric power market design. Edison Electric Institute, 2002. URL http://www.eei.org/industry_issues/retail_services_and_delivery/wi%se_energy_use/demand_response/demandresponserole.pdf.

Adam Coates, Pieter Abbeel, and Andrew Y. Ng. Autonomous helicopter flight using reinforcement learning. In Encyclopedia of Machine Learning, pages 53–61. 2010.

Ahmad Faruqui and Stephen George. Quantifying customer response to dynamic pricing. The Electricity Journal, 18(4):53 – 63, 2005. URL http://www.sciencedirect.com/science/article/B6VSS-4G1YW67-3/2/44466f4%7c4dd993c8f13c290bd91dc97.

Karen Herter. Residential implementation of critical-peak pricing of electricity. Energy Policy, 35(4):2121 – 2130, 2007a. ISSN 0301-4215. doi: DOI:10.1016/j.enpol.2006.06.019. URL http://www.sciencedirect.com/science/article/B6V2W-4KSSWHP-2/2/57823b87c8a8355805b5896909d1f016.
Karen Herter. An exploratory analysis of California residential customer response to critical peak pricing of electricity. *Energy*, 32(1):25 – 34, January 2007b. URL [http://www.sciencedirect.com/science/article/B6V2S-4JG5F91-2/2/bb70d54%6082f9f5483829aabeef5279e](http://www.sciencedirect.com/science/article/B6V2S-4JG5F91-2/2/bb70d54%6082f9f5483829aabeef5279e).

E. Koch and M.A. Piette. Architecture concepts and technical issues for an open, interoperable automated demand response infrastructure. In *Grid Interop Forum*, Albuquerque, NM, US, November 2007.

M. LeMay, R. Nelli, G. Gross, and C. A. Gunter. An integrated architecture for demand response communications and control. In *Proc. of the 41st Hawaii International Conference on System Sciences*, 2008.

D. O'Neill, M. Levorato, A. J. Goldsmith, and U. Mitra. Residential demand response using reinforcement learning. In *IEEE SmartGridComm*, Gaithersburg, Maryland, USA, OCT 2010.

M. A. Piette, O. Sezgen, D. Watson, N. Motegi, C. Shockman, and L. ten Hope. Development and evaluation of fully automated demand response in large facilities. January 2005. URL [http://escholarship.org/uc/item/4r45b9zt](http://escholarship.org/uc/item/4r45b9zt).

M. A. Piette, D. Watson, N. Motegi, and S. Kiliccote. Automated critical peak pricing field tests: 2006 pilot program description and results. In *LBNL Report 62218*, Albuquerque, NM, US, May 2007.

J.G. Roos and I.E. Lane. Industrial power demand response analysis for one-part real-time pricing. *Power Systems, IEEE Transactions on*, 13(1):159 –164, feb 1998.

R.S. Sutton and A.G. Barto. *Reinforcement learning*. MIT Press, 1998.

C. Watkins. *Learning from Delayed Rewards*. PhD thesis, University of Cambridge, 1989.
APPENDICES

Appendix A  Bellman Equation

It is worth pointing out that if the transition model of the device based MDP and the dissatisfaction function of the consumer are known, the device based MDP can be solved by dynamic programming (DP). Following ideas in classical DP, in this section, we derive the Bellman equation from which we can compute the optimal Q-function.

Recall that $x_t = [P_t, s_t, g_t]^T$ and the action space at each state is $A = \{\text{off}, \text{on}\}$, we have

- If the smart device has not received a consumer request in the current episode, we have
  \[
  Q^*(x_t, \text{on}) = P_t C + \gamma \tilde{U}_r (s_t) + \alpha \mathbb{E}\left\{ \min_{a \in A} Q^* \left( [P_{t+1}, s_{t+1}, g_{t+1}]^T, a \right) \right\},
  \]
  where the expectation is over $P_{t+1}$, $s_{t+1}$ and $g_{t+1}$. Specifically, $P_{t+1}$ is drawn according to the transition probability of the price Markov chain, and $[s_{t+1}, g_{t+1}]^T$ is drawn according to the “device portion” regeneration distribution $\pi_0$.

  On the other hand, we have
  \[
  Q^*(x_t, \text{off}) = \alpha \mathbb{E}\left[ \sum_{s'=-W}^{g_{\max}} \sum_{g=1}^{g_{\max}} p_{s_t, s', g} \min_{a \in A} Q^* \left( [P_{t+1}, s', g]^T, a \right) + (1 - p_{s_t}) \min_{a \in A} Q^* \left( [P_{t+1}, s(t+1), 0]^T, a \right) \right],
  \]
  where the expectation is over $P_{t+1}$. Specifically, $p_{s_t} = \sum_{s'=-W}^{0} \sum_{g=1}^{g_{\max}} p_{s_t, s', g}$ is the probability that a consumer request will be received in the next time step, $s(t+1) = s_t + 1$ if $s_t < \hat{W}$ and $s(t+1) = \hat{W}$ if $s_t = \hat{W}$, and $P_{t+1}$ is drawn according to the transition probability of the price Markov chain.

- If the smart device has already received a consumer request in the current episode, we have
  \[
  Q^*(x_t, \text{on}) = P_t C + \gamma \tilde{U}_r (s_t, g_t) + \alpha \mathbb{E}\left\{ \min_{a \in A} Q^* \left( [P_{t+1}, s_{t+1}, g_{t+1}]^T, a \right) \right\},
  \]
  where the expectation is over $P_{t+1}$, $s_{t+1}$ and $g_{t+1}$. Similarly, $P_{t+1}$ is drawn according to the transition probability of the price Markov chain, and $[s_{t+1}, g_{t+1}]^T$ is drawn according to the “device portion” regeneration distribution $\pi_0$.

  On the other hand, we have
  \[
  Q^*(x_t, \text{off}) = \alpha \hat{p}_{s_t, g_t} \mathbb{E}\left[ \min_{a \in A} Q^* \left( [P_{t+1}, s_t + 1, g_t]^T, a \right) \right] + (1 - \hat{p}_{s_t, g_t}) \left[ \gamma \tilde{U}_c (s_t, g_t) + \alpha \mathbb{E}\left\{ \min_{a \in A} Q^* \left( [P_{t+1}, s_{t+1}, g_{t+1}]^T, a \right) \right\} \right].
  \]
  Note that $\hat{p}_{s_t, g_t} = 0$ if $s_t = \hat{W}$. The expectation in the first line is over $P_{t+1}$, where $P_{t+1}$ is drawn according to the transition probability of the price Markov chain. On the other hand, the expectation in the third line is over $P_{t+1}$, $s_{t+1}$ and $g_{t+1}$, where $P_{t+1}$ is also drawn according to the transition probability of the price Markov chain, and $[s_{t+1}, g_{t+1}]^T$ is drawn according to the “device portion” regeneration distribution $\pi_0$. 

24
From the classical DP theory, the optimal Q-function $Q^*$ is the unique solution of the above-derived Bellman equation. Furthermore, many DP algorithms, such as value iteration and policy iteration, can be used to compute $Q^*$. We observe that for many DP algorithms, computing $Q^*$ is tractable since the cardinalities of the state space $S$ and action space $A$ in a device based MDP are usually small. Once $Q^*$ is available, one optimal policy $\mu^*$ is

$$
\mu^*(x_t) \in \arg\min_{a \in A} Q^*(x_t, a).
$$

**Appendix B  Proof for Theorem 1**

**Proof for Theorem 1:**

Note that $\forall \mu : S \times A \to [0, 1], V_\mu$ can be expressed as

$$
V_\mu = A_\mu + \gamma B_\mu,
$$

where $A_\mu$ is the expected infinite-horizon discounted electricity bill, and $B_\mu$ is the expected infinite-horizon discounted user dissatisfaction on rescheduling. Furthermore, neither $A_\mu$ nor $B_\mu$ depends on $\gamma$.

First, we prove that $V_{\mu_{\text{base}}}$ does not depend on $\gamma$. Note that under policy $\mu_{\text{base}}$, the EMS will never initiate a job; furthermore, any job initiated by the user is either completed at its target time or canceled by the user before its target time. Thus, under Assumption 5, we have $B_{\mu_{\text{base}}} = 0$. So we have $V_{\mu_{\text{base}}} = A_{\mu_{\text{base}}}$, which does not depend on $\gamma$.

Second, we prove that there exists a $\gamma^* > 0$ s.t. $\mu_{\text{base}}$ is an optimal policy when $\gamma > \gamma^*$. Note that under Assumption 5, $\mu_{\text{base}}$, which is a deterministic policy, minimizes $B_\mu$, since $B_\mu \geq 0$ for any $\mu$ and $B_{\mu_{\text{base}}} = 0$. Note that there are finite deterministic policies, and we use $\Delta B$ to denote the “second best” $B_\mu$ under deterministic policies

$$
\Delta B = \min_{\text{deterministic } \mu : B_\mu > 0} B_\mu.
$$

Note that the maximum cost reduction from $A_\mu$ is $\frac{1}{1-\alpha} (P_{\text{max}} - P_{\text{min}}) C$, where $P_{\text{max}} = \max_{P \in \mathcal{P}} P$ is the highest energy price and $P_{\text{min}} = \min_{P \in \mathcal{P}} P$ is the lowest energy price. Thus, if we choose

$$
\gamma^* = \frac{(P_{\text{max}} - P_{\text{min}}) C}{(1-\alpha) \Delta B},
$$

then, $\forall \gamma > \gamma^*$, $\mu_{\text{base}}$ outperforms all the deterministic policies (i.e. $V_{\mu_{\text{base}}} \leq V_\mu$ for any deterministic $\mu$). Consequently, it is an optimal policy. Thus $\text{DRP} = 0$ for any $\gamma > \gamma^*$. As a result, we have $\text{DRP} \to 0$ as $\gamma \to \infty$.

Third, we prove that $\text{DRP}$ is a non-increasing function of $\gamma$. To formalize the result, we use $V_\mu(\gamma)$ to denote the performance of policy $\mu$ at tradeoff parameter $\gamma$. Recall $V_\mu(\gamma) = A_\mu + \gamma B_\mu$, and $B_\mu \geq 0$ under Assumption 5, thus $V_\mu(\gamma)$ is a non-decreasing function of $\gamma$ for any $\mu$. Furthermore, we use $\mu^*(\gamma)$ to denote an optimal policy in the problem instance with tradeoff parameter $\gamma$. Thus, for any $0 \leq \gamma_1 \leq \gamma_2$, we have that

$$
V_{\mu^*(\gamma_1)}(\gamma_1) \leq V_{\mu^*(\gamma_2)}(\gamma_1),
$$

since $\mu^*(\gamma_1)$ is an optimal policy in the problem instance with parameter $\gamma_1$, and

$$
V_{\mu^*(\gamma_2)}(\gamma_1) \leq V_{\mu^*(\gamma_2)}(\gamma_2),
$$
since $\mu^*(\gamma_2)$ is a fixed policy and $\gamma_1 \leq \gamma_2$. Thus, $V_{\mu^*}(\gamma_1)(\gamma_1) \leq V_{\mu^*}(\gamma_2)(\gamma_2)$, that is, $V_{\mu^*}$ is a non-decreasing function of $\gamma$. Since $\text{DRP} = V_{\mu^{\text{base}}} - V_{\mu^*}$, and $V_{\mu^{\text{base}}}$ does not depend on $\gamma$, and $V_{\mu^*}$ is non-decreasing in $\gamma$, thus, $\text{DRP}$ is non-increasing in $\gamma$. From the definition of $\text{RDRP}$, it is also non-increasing in $\gamma$.

Finally, we prove that $\text{RDRP} = 1$ when $\gamma = 0$. Note when $\gamma = 0$, the EMS does not care about the user’s dis-satisfaction on job rescheduling. Consider a policy $\mu'$ under which

- The EMS will never initiate a job.
- The EMS ignores all the jobs initiated by the user; it just waits for the user to cancel the requested job.

Obviously, when $\gamma = 0$, we have $V_{\mu'} = 0$. Thus we have $V_{\mu^*} \leq V_{\mu'} = 0$. On the other hand, from Assumption 5, we have $V_{\mu^*} \geq 0$ (since the energy price is always strictly positive and and the user’s dissatisfaction function is always non-negative). Thus we have $V_{\mu^*} = 0$ and

$$\text{RDRP} = \frac{V_{\mu^{\text{base}}} - V_{\mu^*}}{V_{\mu^{\text{base}}}} = 1.$$

Q.E.D.

**Appendix C Possible Extensions**

We discuss some possible extensions of the device based MDP model proposed in the paper. It is worth pointing out that many assumptions on this model are proposed to simplify the exposition. We can easily relax some of these assumptions to obtain a more general device based MDP model.

We now discuss two relaxations that are of particular interest: stacking request and dependent devices.

**Stacking request**

Under the current device based MDP model, the smart device will ignore a new user request if it currently has an uncompleted request. If the user wants to replace the existent uncompleted request with a new request, they must cancel the existent request first, and then send the new request to the device.

However, in some cases, the user might want to start a new request while keeping the existent uncompleted request. For instance, a user sent a request with target time 4PM to the AC at 1PM; at 2PM, this request was still uncompleted, and the user wanted to send a new request with target time 5PM to the AC while keeping the old request. Starting a new request without canceling the old request is referred to as initiating a stacking request.

We can easily extend our proposed device based MDP model to take care of the stacking request by allowing $\tau_g$, a request target time, to be a vector. Specifically, in the extended MDP model, the user can request multiple jobs in a request $J = (n, \tau_r, \tau_g, g)$, where $\tau_g$ is the target time vector and its components denote when the user prefers each of their requested jobs to be completed. Requested device $n$, request time $\tau_r$ and priority $g$ are defined the same as in Section II. With the new definition of $\tau_g$, a stacking request can be realized as follows: assume the target time vector of the current user request is $\tau_g$, and the user wants to stack new jobs with target time vector $\tau_g'$ on it, then they can cancel the existent request first, and then start a new request with target time vector $[\tau_g, \tau_g']^T$. 

26
Obviously, allowing $\tau_g$ to be a vector will make the device based MDP model more complicated; however, it does not fundamentally change the problem and all the existent results still apply after proper modifications. For computational tractability, one should also upper bound the length of the target time vector.

**Dependent devices**

In Assumption 3, we assume that consumer requests/cancellations to different devices are statistically independent. As we have discussed after Assumption 3, it should be relaxed for some devices: specifically, devices can be classified into categories, and requests to devices in the same category (such as laundry machine and dryer) are statistically dependent.

One way to overcome this problem is to combine all the devices in the same category as a “super device”. After such combinations, the requests/cancellations to different “super devices” can be assumed to be statistically independent. Obviously, allowing “super devices” will make the device based MDP model more complicated; however, it also does not fundamentally change the problem and all the existent results still apply after proper modifications.