Adaptive Demand Response: Online Learning of Restless and Controlled Bandits

Qingsi Wang, Mingyan Liu, Johanna L. Mathieu
University of Michigan, Ann Arbor

Abstract—**The capabilities of electric loads participating in load curtailment programs are often unknown until the loads have been told to curtail (i.e., deployed) and observed. In programs in which payments are made each time a load is deployed, we aim to pick the "best" loads to deploy in each time step. Our choice is a tradeoff between exploration and exploitation, i.e., curtailing poorly characterized loads in order to better characterize them in the hope of benefiting in the future versus curtailing well-characterized loads so that we benefit now. We formulate this problem as a multi-armed restless bandit problem with controlled bandits. In contrast to past work that has assumed all load parameters are known allowing the use of optimization approaches, we assume the parameters of the controlled system are unknown and develop an online learning approach. Our problem has two features not commonly addressed in the bandit literature: the arms/processes evolve according to different probabilistic laws depending on the control, and the reward/feedback observed by the decision-maker is the total realized curtailment, not the curtailment of each load. We develop an adaptive demand response learning algorithm and an extended version that works with aggregate feedback, both aimed at approximating the Whittle index policy. We show numerically that the regret of our algorithms with respect to the Whittle index policy is of logarithmic order in time, and significantly outperforms standard learning algorithms like UCB1.**

I. Introduction

Electric loads participating in demand response programs provide a variety of benefits to electric power systems including increased power system reliability and power market efficiency [1]. However, the responses of loads to curtailment signals are uncertain [2] because load behavior is complex and influenced by a variety of stochastic factors including weather and human behavior. Generally, detailed load models are unavailable and we do not have full access to realtime information about load models, states, or disturbances due to limited communications. Subsequently, a load's ability to curtail is often only known after it has been told to curtail (i.e., deployed) and observed. This results in a tradeoff between exploration and exploitation [3], i.e., pursuing potential gain from poorly characterized loads so as to improve our characterization which hopefully leads to future gains versus harvesting immediate benefits from well-characterized loads.

In this paper, we formulate load curtailment as a multi-armed restless bandit problem, where a decision maker must repeatedly select multiple arms/processes from a set, generating state-dependent rewards unknown a priori. The system is "restless" because the states evolve regardless of the control. Here the decision maker is a load aggregator (e.g., a utility company or third party curtailment service provider) with a fixed budget. We assume the aggregator must pay a load each time it is deployed and therefore selects only a portion of the loads for deployment in each time step. The aggregator's goal is to deploy the "best" loads which allow her to maximize load curtailment subject to her budget.

There has been a considerable amount of research on bandit problems over the past few decades along two directions. The first class of problems, referred to as the optimization version of the bandit problem within this paper, concerns the case where the reward process associated with each arm is described by a well-defined probabilistic model. In this case, the problem is in essence a stochastic control problem (e.g., a Markov Decision Process), and the objective is to derive a sequential decision process that maximizes a total reward (average or discounted) over a finite or infinite horizon. Index policies have often been used as solutions to bandit problems in this category. Specifically, each arm is associated with a (scalar) index and the arms are selected in a greedy fashion with respect to their indices. Seminal results include the Gittins index [4] for rested bandits where the state of an arm remains static when not selected, and Whittle's heuristic index [5] for restless bandits where the state of an arm continues to evolve regardless of the control.

The second class of problems, referred to as the learning version of the bandit problem within this paper, concerns the case where the reward process is unknown. This further splits into two cases: In the first, the process is assumed to follow a certain probabilistic model with unknown parameters, e.g., a Markov chain with unknown transition probabilities or an i.i.d. process with unknown probability distribution. This is commonly referred to as the stochastic bandit, see e.g., [6]. In the second, the reward process is assumed arbitrary, i.e., no probabilistic structure is imposed, which is often used to capture an adversarial process. This is commonly referred to as the non-stochastic bandit, see e.g., [7]. The performance of a learning policy is typically measured by the *regret*, defined as the gap between the rewards obtained by the given policy and that of a reference policy (typically one used by a genie/oracle).

Under the optimization version, [3] formulated demand response as a multi-armed restless bandit problem, assuming a two-state Markovian model, and calculated the Whittle index policy. It was shown to outperform a naive greedy policy by 5–10%. Prior knowledge of the state transition probabilities, which are assumed different for the uncontrolled and controlled system, is key to deriving the indices. The two-state model sufficiently approximates the dynamic response capability of a variety of types of loads and is analytically tractable; however, in a realistic system with highly heterogeneous loads it would be difficult to obtain all necessary load parameters, especially the parameters associated with the controlled response.

Here, we relax assumptions on knowledge of the system parameters by applying the learning version of the multi-armed restless bandit problem. Specifically, we assume the dynamic behavior of loads can be modeled with a set of two-state Markov chains as in [3], but unlike [3] we assume the transition probabilities of the controlled system are unknown a priori. Therefore, we solve a stochastic bandit problem.

Past work on multi-armed bandit learning algorithms has mainly focused on reward processes that are *uncontrolled* Markov chains, meaning that the control decision does not affect the underlying state transitions. In other words, each reward process is governed by a single set of transition probabilities, so the state transitions are independent of the decision maker's actions. This setting most aptly captures problems in which the selection of an arm does not physically alter that arm. Examples include the celebrated UCB1 algorithm [8] and its derivatives (e.g., [9], [10]), which are concerned with uncontrolled i.i.d. processes, and [6] and [11], which focus on uncontrolled finite-state Markov chains. A notable exception is [12], which considered a controlled Markov chain; however, in this study there is only one arm/process and consequently its states are always perfectly observed. Past work also heavily relies on obtaining observations (of the state or the reward) from each of the selected/activated arms, even if multiple are activated at a time. This allows us to learn each arm directly and separately. However, the demand response problem presents two challenges: 1) load are governed by different statistics when the system is controlled and uncontrolled [3], and 2) individual feedback from each deployed load is often unobservable in a practical system; the feedback is instead of the form of a (noisy) aggregate load curtailment. These challenges render existing multi-armed bandit learning algorithms inapplicable to the demand response problem.

In light of the above discussion, we seek a solution to the demand response problem by formulating it as a multi-armed restless bandit learning problem for controlled Markov processes with noisy aggregate feedback. We propose an efficient online learning algorithm that is adaptive to the dynamic response capability of each load and empirically reliable for a broad class of loads. The remainder of this paper is organized as follows. We present our system model and the proposed learning algorithm in Section II and III. We numerically evaluate the performance of this algorithm in Section IV and Section V concludes.

II. System Model and Preliminaries

A. Model

Consider a system that consists of N electric loads that can be deployed by an aggregator, indexed by $[N] = \{1, 2, \dots, N\}$. The dynamics of a load are illustrated in Fig. 1 and described by a pair of two-state Markov chains, one characterizing state transitions when the load is active (i.e., deployed) and one characterizing state transitions when the load is passive (i.e., not deployed). In each case, the load may be in one of two states, available for load curtailment or unavailable. For example, a refrigerator is available if it is powered on and within its temperature limits, and unavailable if it is powered off or outside its temperature limits.

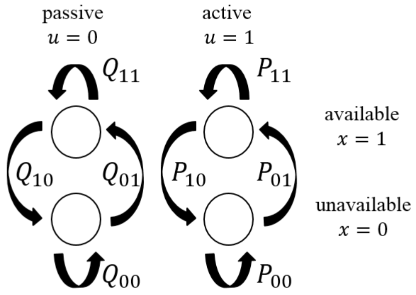


Fig. 1. A two-state Markov chain model representing a load's availability for load curtailment, as in [3].

Formally, we will denote a load by a controlled time-homogeneous Markov chain $\{X_k(t)\}_{t=1}^{\infty}$, $\forall k \in [N]$, where $X_k(t) \in \{0, 1\}$. The *curtailment capacity* (in units of power) of load k is given by a constant $c_k > 0$ whenever it is available. Let $U_k(t)$ be the control of load k by the aggregator at time t with $U_k(t) = 1$ if load k is selected and 0 otherwise. The corresponding Markov chains induced by the control actions are given by

$$\mathbb{P}(X_k(t+1) = j | X_k(t) = i, U_k(t) = 1) = P_{ij}^k,$$
$$\mathbb{P}(X_k(t+1) = j | X_k(t) = i, U_k(t) = 0) = Q_{ij}^k,$$

where $P_k = [P_{ij}^k]_{2\times 2}$ is the transition matrix under deployment or the *active transition matrix*, and $Q_k = [Q_{ij}^k]_{2\times 2}$ the *passive transition matrix*, as shown in Fig. 1. To put in the context of the bandit problem framework, a load maps to an arm, and the deployment of a load maps to the activation or playing of an arm.

We assume that the passive matrices Q_k are known to the aggregator, while the active matrices P_k are *unknown*. Such an assumption is justified for the following reasons. A model for the uncontrolled behavior of the load could be determined based on the type of load and manufacturer's specifications. When a load signs up to participate in a demand response program it could be required to provide this information to the load aggregator. However, it is unreasonable to assume that we would have a good model of the controlled behavior of the load. The best way to build such a model is to observe the load's response to curtailment signals over time. This could be done before the load begins to participate in the program, or, as proposed here, while it is participating in the program. In the latter, we must use online learning.

The decision making of the aggregator is sequential, and the decision epochs are given by discrete time slots $t = 1, 2, \dots$. In each slot, the aggregator chooses up to K loads to participate in demand response, where $K < N$.

The aggregator can only obtain feedback/measurements from deployed loads. In particular, this feedback takes an aggregated form in practice especially when K and N are large, i.e., the aggregator only gets to observe the total amount of realized curtailment, but not the amount of curtailment achieved for each load. This results in significant challenge to the design of a learning algorithm which typically tries to estimate the quality and dynamics of each individual process/arm. In this paper, we will first present a learning algorithm assuming the simpler case, whereby the aggregator can observe individual states on each activated load (Section III-A). We then extend our algorithm to the case when only noisy aggregate measurements of curtailed capacity are available to the aggregator (Section III-B).

The control actions $U_k(t)$ for all $k \in [N]$ are determined by a control rule g_t at time t, based on all past observations and control actions. The collection of control rules over time $g = (g_1, g_2, \ldots)$ will be called a policy in this paper. The objective of the aggregator is to maximize the expected discounted infinite-horizon kW capacity

$$\mathbb{E}^g \left\{ \sum_{t=0}^{\infty} \alpha^t \sum_{k=1}^{N} c_k U_k(t) X_k(t) \right\},$$

subject to the constraint $\sum_{k=1}^{N} U_k(t) = K$ for all t, where $0 < \alpha < 1$ is the discounting factor and we use the superscript g to emphasize the dependence of the expectation operator on the policy g. This objective function can be used to maximize curtailment of demand responsive loads during system peaks, which is a common objective in traditional demand response programs, e.g., [13].

If in addition to Q_k, we also know P_k for each $k \in [N]$, then the problem would reduce to the well-studied restless bandit problem of the optimization version. While a general structured solution to this type of problems remains elusive (and the hardness of this problem has been shown to be PSPACE-complete [14]), various heuristic policies have been proposed in the literature for specific problems. These often take the form of index policies, where a scalar index is computed at each step for each arm using only statistics of that arm, and arm(s) with the highest index (indices) are selected for play. Of particular interest is the Whittle index policy, which is suboptimal in general, but optimal under the relaxation $\mathbb{E}\{\sum_{k=1}^{N} U_k(t)\} = K$, i.e., requiring on average K arms are played at each time rather than exactly K arms are played at each time. This is widely used as an efficient heuristic solution in many problem instances.

Below we give a brief review of the Whittle index policy derived in [3] for the above problem assuming both P_k and Q_k are known. Then in Section III we present two learning algorithms, one in the case of explicit feedback from each active load and one in the case of aggregate feedback, that attempt to track the performance of this policy by estimating the unknown matrices P_k and computing an approximate version of this policy.

B. The Whittle index policy and the regret measure

Let $\pi_k(t)$ be the probability that $X_k(t) = 1$ given all past observations and control actions, i.e., the *belief state* at time t, which is a sufficient statistic for optimal control [15]. The value of the aggregator's objective can then be written as

$$\mathbb{E}^g \left\{ \sum_{t=0}^{\infty} \alpha^t \sum_{k=1}^{N} c_k U_k(t) \pi_k(t) \right\}.$$

The evolution of the belief state is given by

$$\pi_k(t+1) = \begin{cases} P_{01}^k, & \text{if } X_k(t) = 0, U_k(t) = 1 \\ P_{11}^k, & \text{if } X_k(t) = 1, U_k(t) = 1 \\ \phi_k \pi_k(t), & \text{if } U_k(t) = 0 \end{cases} \qquad (1)$$

where the operator ϕ_k evolves the belief state when load k is passive, i.e.,

$$\phi_k \pi_k(t) = Q_{11}^k \pi_k(t) + Q_{01}^k (1 - \pi_k(t)).$$

Under certain order conditions on the transition probabilities, [3] derived the Whittle index w_k, which can be computed with extremely low complexity and is detailed in Appendix A. The Whittle index policy can be then summarized as follows, assuming known transition probability matrices.

```
Whittle Index Policy
Initialization: Set the belief state $\pi_k = 1/2$ for all $k \in [N]$
For time $t = 1, 2, \ldots, T$ do:
    1) Compute the Whittle index $w_k$ for each load $k$.
    2) Dispatch the $K$ loads with largest indices.
    3) Observe the states of the active (deployed) loads, and
       update the belief state $\pi_k$ as in (1) for all loads.
```

The computation of the Whittle indices explicitly relies on the complete knowledge of the active and the passive transition matrices (see the Appendix A). In the case when the active transition matrices P_k are unknown, which we consider in this paper, we measure the performance of a given policy g with respect to the Whittle index policy with complete knowledge, denoted by g_W. In particular, we use the notion of regret, which is given by the gap between the total *undiscounted* kW capacity that can be curtailed by g_W and that by g. Formally, let

$$G(g, T) = \mathbb{E}^g \left\{ \sum_{t=0}^{T} \sum_{k=1}^{N} c_k U_k(t) X_k(t) \right\},$$

and we then define the regret $R(g, T)$ of the policy g by

$$R(g, T) = G(g_W, T) - G(g, T).$$

We consider a policy as efficient if the regret is *sublinear* over time, i.e., $R(g, T) = o(T)$, or so-called *no regret* for the average regret. A logarithmic growth rate of the regret is typically order optimal in the context of learning uncontrolled processes with probabilistic models[1], with respect to various baseline policies (e.g., a static policy and the resulting regret is often called the weak regret, or the optimal dynamic policy that gives rise to the notion of strong regret).

In the next section, we present a policy that learns the active transition matrices when they are unknown over time and *mimics* or tracks the Whittle index policy with estimated parameters. We show in Section IV that the proposed algorithm empirically exhibits logarithmic regret; formally establishing this result is part of our ongoing work.

[1] For arbitrary but bounded processes associated with each arm, the order optimal regret is in general affine in the square root of time, see e.g. [7].

III. Adaptive Demand Response Algorithms

A. Learning with individual load observations

Our basic learning algorithm, referred to as the Adaptive Demand Response Learning Algorithm (ADRLA), works under the assumption that individual load feedback upon deployment is available. This is then extended to the case where only aggregate feedback is available in the next subsection. ADRLA is similar in structure to the well-known ε-greedy algorithm (see e.g. [8]) and works as follows. The discrete time slots are divided into blocks of equal length $2\lceil N/K \rceil$. Each block is either an exploitation block or an exploration block, with the probability of being the latter diminishing inversely proportional to the index of the block. Over time the algorithm computes an estimated active matrix $\hat{P}_k$ for each load k. In an exploitation block, the algorithm uses the estimates $\hat{P}_k$ to compute the Whittle indices and plays those with the highest indices. In an exploration block, the algorithm improves the estimates $\hat{P}_k$: it sequentially samples loads in batches of size K (except possibly the last batch if N is not a multiple of K) for two consecutive slots, so as to observe an active state transition. It maintains a matrix C_k of counters, of which the (i, j) entry C^k_{ij} is the total number of active transitions observed from state i to j in exploration, $i, j \in \{0, 1\}$. The estimate $\hat{P}_k$ is then formed by

$$\hat{P}^k_{ij} = \frac{C^k_{ij}}{C^k_{i0} + C^k_{i1}} \ . \tag{2}$$

In other words, ADRLA maintains the sample mean estimates of the transition probabilities in the active transition matrices. The ADRLA algorithm is detailed as follows. Here the constants L and ε_b control the rate of learning/exploration. There typically needs to be a lower bound on the value of L as a sufficient condition for sub-linear regret, while the choice of a diminishing ε_b, inversely proportional to time, may lead to logarithmic regret as it does for sample mean-based algorithms like UCB1 [8] or RCA [8].

```
Adaptive Demand Response Learning Algorithm
Parameter: A constant $L > 0$.
Initialization (for all $k$):
  1) Set the initial belief state $\pi_k = 1/2$.
  2) Set the counter $C_k$ of active transitions by $C^k_{ij} = 1$ for all
     entries, and form $\hat{P}_k$ as in (2) for all $i$.
  3) Define a sequence $\varepsilon_b$ by $\varepsilon_b = \min\{1, \frac{L}{b}\}$ for all $b \in \mathbb{N}$.
For block $b = 1, 2, \dots, n$ do
  - With probability $1 - \varepsilon_b$ enter an EXPLOIT block and with
    probability $\varepsilon_b$ enter an EXPLORE block.
  - EXPLOIT:
        For $\ell = 1, 2, \dots, 2\lceil N/K \rceil$ do
        1) Compute the Whittle index $w_k$ for each load $k$ using
           $\hat{P}_k$ and $Q_k$.
        2) Dispatch the $K$ loads with the largest indices.
        3) Observe the states of the active loads, and update
           $\pi_k$ as in (1) using $\hat{P}_k$ and $Q_k$ for all loads.
  - EXPLORE:
        For $\ell = 1, 2, \dots, \lceil N/K \rceil$ do
        1) Dispatch loads from $k = (\ell - 1)K + 1$ to
           $\min\{\ell K, N\}$ for two consecutive slots and observe
           states $i^1_k$ and $i^2_k$ for each active load $k$.
        2) Update $\pi_k$ after observing $i^1_k$ in the first slot.
           Increase the transition counter $C^k_{i^1_k i^2_k}$ by one after
           observing $i^2_k$ and update $\hat{P}_k$ for each active load $k$.
           Update $\pi_k$ for the second slot.
```

ADRLA is used in this paper as a heuristic solution; its regret performance is empirically examined in the next section. A more rigorous treatment of its regret performance is part of our ongoing work. Also note that the operation of ADRLA does not rely on the stationarity of load dynamics, and is applicable when the response capability of loads are given by controlled *nonhomogeneous* Markov chains.

B. Learning with noisy aggregate curtailment measurement

In this subsection, we extend ADRLA to the more realistic scenario in which only noisy aggregate curtailment is measured and observed by the demand response aggregator. This extension will be referred to as the ADRLA-A algorithm. In this case, instead of observing the individual state X_k of each active load, the aggregator is only given the feedback $Y(t) = \sum_{k=1}^{N} c_k U_k(t) X_k(t) + Z(t)$ at time t, where $Z(t)$ is an observation noise. We adopt the Bayesian inference framework proposed in [3], which we briefly describe as follows. Assume that the distribution of $Y(t)$, conditional on the total curtailed capacity $\sum_k c_k U_k(t) X_k(t) = \sum_k c_k u_k x_k$, is given by $f(y; \sum_i c_k u_k x_k)$, where $x^u = (x_k, k : u_k = 1)^\top \in \{0, 1\}^K$ denotes a particular realization of the states of the active loads. Let $p(x^u)$ be the prior distribution of the states. Then the posterior distribution of the states of active loads is given by

$$p(x^u|y) = \frac{f(y; \sum_k c_k x_k u_k) p(x^u)}{p(y)}$$

where $p(y) = \sum_{x^u} f(y; \sum_k c_k u_k x_k) p(x^u)$. Consequently, the marginal distribution of an active load k is

$$p_k(i|y) = \sum_{x^u : x_k = i} p(x^u|y),$$

where $i \in \{0, 1\}$, and accordingly the belief state evolves as

$$\pi_k(t+1) = \begin{cases} P^k_{01} p_k(0|y) + P^k_{11} p_k(1|y), & U_k(t) = 1 \\ \phi_k \pi_k(t), & U_k(t) = 0 \ , \end{cases} \tag{3}$$

given the aggregate measurement y. The computation of the marginal distribution involves enumerating exponentially many values of the state vector x, but low-complexity approximations are available in [3], which are detailed in Appendix B. The ADRLA-A algorithm consists of adapting ADRLA such that the belief state is now updated using (3) after observing the noisy aggregate measurement y.

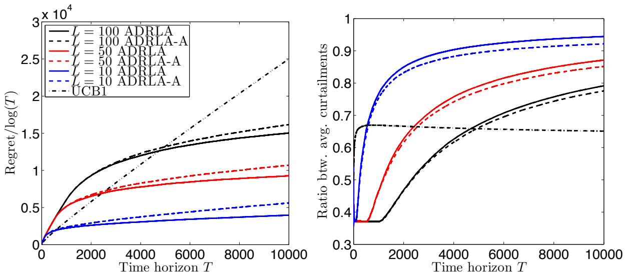
IV. Numerical Results

In this section, we report the numerical results on the performance of the proposed ADRLA and ADRLA-A algorithms. We consider 1,000 loads, and in each time slot, the aggregator can deploy 200 loads.

We first conduct two sets of numerical tests considering two different load populations:

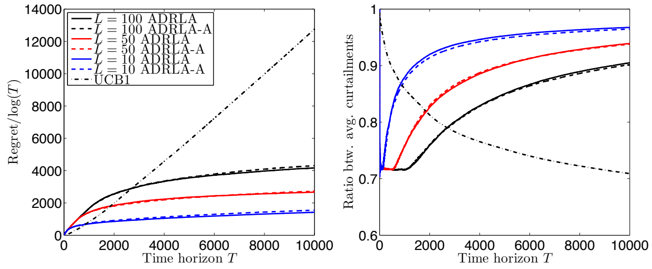
1) heterogeneous loads: the transition matrices of each load are independently and randomly generated, and
2) homogeneous loads: all loads share the same set of transition matrices, which are randomly generated.

All the numeric value of parameters are generated subject to the order conditions in Appendix A. The performance metrics that we consider include the regret and the ratio between the average curtailment capacities of ADRLA or ADRLA-A and the Whittle index policy. Recall that the belief states are updated using (1) in ADRLA, and (3) in ADRLA-A. When the feedback is in aggregate, we assume that the measurement has a normal distribution with the true aggregate curtailment capacity as mean and K as variance, which is known to ADRLA-A for the Bayesian inference. For each set of tests, 100 sample paths are generated to produce empirical mean values of performance metrics. Moreover, we also consider the comparison with an efficient online learning algorithm for uncontrolled processes, and we report the results for the multiple-play version of the UCB1 algorithm [16] (see Appendix C), assuming individual measurements are available when using UCB1. The multi-play UCB1 is theoretically shown to have a sublinear regret for i.i.d. processes with respect to the optimal static policy[2] under appropriate parameter setup. The baseline policy is always the Whittle index policy described in Section II with full knowledge of transition matrices and individual feedback. Our results are summarized in Figs. 2 and 3.



(a) Regret over time (b) Ratio b/w. the avg. curtailment

Fig. 2. Policy performance for heterogenous loads.

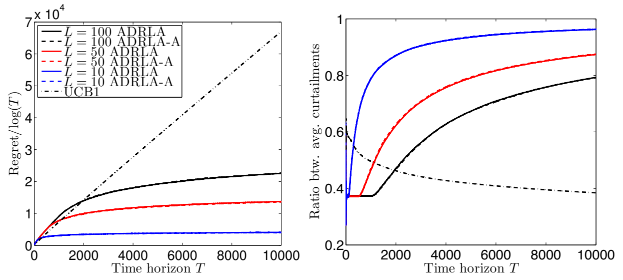


(a) Regret over time (b) Ratio b/w. the avg. curtailment

Fig. 3. Policy performance for homogeneous loads.

As can be seen, the proposed adaptive learning algorithms empirically result in near logarithmic regrets (the regret over the logarithm of time tends to be upper bounded by a constant), and the average curtailment capacity can achieve at least 78% of that of the Whittle index policy by the time horizon we set (an applicable value of the horizon may vary depending on the applications). The UCB1 algorithm is designed for learning uncontrolled processes and its behavior is in general unpredictable with controlled dynamics. In our experiment, though its initial behavior depends on the setup of parameters, the performance of UCB1 is eventually much inferior with the learning rate learning rate parameter $L = 2$ (see Appendix C) than ADRLA/ADRLA-A, and similar performance gap is observed for other values of L in UCB1. We also make a cautious remark that ADRLA/ADRLA-A do not "converge" in the sense that exploration will be performed infinitely often as the time horizon extends, but the total time spent in exploration is only logarithmic of the horizon as how the sequence ε_b is chosen.

We further reduce the range of parameters in the transition matrices, and target the situation that would be more commonly expected in demand response programs. In particular, we consider the case in which active loads are much less available than passive loads. For example, the parameters can be constrained as $Q^k_{11} \geq 0.5$, $0.4 \leq Q^k_{01} \leq Q^k_{11}$, $P^k_{11} \leq 0.1$, and P^k_{11} and P^k_{01} further satisfy the order conditions (see Appendix A). We report the results in Fig. 4 for the previous example with heterogenous loads (i.e., the value of parameters are randomly generated for each load but consistent with the constraints). ADRLA and ADRLA-A are much more similar in performance in this case, and other observation is similar to the previous tests. Similar results can be observed for homogeneous loads in this setup, which are omitted due to the space limit.



(a) Regret over time (b) Ratio b/w. the avg. curtailment

Fig. 4. Policy performance for heterogenous loads that are much less available when active, than passive.

V. Concluding Remarks

Demand response aggregators interact with large numbers of uncertain electric loads and must make deployment decisions without full information. There is a great need for simple, scalable control policies that can handle uncertainty and partial information. The algorithms presented here are heuristics but due to their simplicity they are powerful tools, allowing aggregators to learn about loads even as they benefit from adaptive demand response.

References

[1] DOE, "Benefits of demand response in electricity markets and recommendations for achieving them," tech. rep., Department of Energy Report to the US Congress, 2006.

[2] Note that for i.i.d. processes, the optimal dynamic and static policies are equivalent, i.e., always pulling the arms with the greatest mean values.

[2] J. Mathieu, D. Callaway, and S. Kiliccote, “Variability in automated responses of commercial buildings and industrial facilities to dynamic electricity prices,” *Energy and Buildings*, vol. 43, pp. 3322–3330, 2011.

[3] J. Taylor and J. Mathieu, “Index Policies for Demand Response,” *IEEE Transactions on Power Systems*, vol. 29, pp. 1287–1295, May 2014.

[4] J. C. Gittins, “Bandit Processes and Dynamic Allocation Indices,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 41, no. 2, pp. pp. 148–177, 1979.

[5] P. Whittle, “Restless Bandits: Activity Allocation in a Changing World,” *Journal of Applied Probability*, vol. 25, pp. pp. 287–298, 1988.

[6] C. Tekin and M. Liu, “Online Learning of Rested and Restless Bandits,” *Information Theory, IEEE Transactions on*, vol. 58, pp. 5588–5611, Aug 2012.

[7] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, “The Nonstochastic Multiarmed Bandit Problem,” *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, 2003.

[8] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time Analysis of the Multiarmed Bandit Problem,” *Mach. Learn.*, vol. 47, no. 2-3, pp. 235–256, 2002.

[9] P. Auer and R. Ortner, “UCB Revisited: Improved Regret Bounds for the Stochastic Multi-Armed Bandit Problem,” *Periodica Mathematica Hungarica*, vol. 61, no. 1-2, pp. 55–65, 2010.

[10] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, “Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting,” *IEEE Transactions on Information Theory*, vol. 58, pp. 3250–3265, May 2012.

[11] C. Tekin and M. Liu, “”Learning of Uncontrolled Restless Bandits with Logarithmic Strong Regret,” *IEEE Transactions on Information Theory*, under review.

[12] R. Agrawal, D. Teneketzis, and V. Anantharam, “Asymptotically Efficient Adaptive Allocation Schemes for Controlled Markov Chains: Finite Parameter Space,” *Automatic Control, IEEE Transactions on*, vol. 34, pp. 1249–1259, Dec 1989.

[13] PG&E, “Peak Day Pricing.” Pacific Gas and Electric Company, 2014.

[14] C. H. Papadimitriou and J. N. Tsitsiklis, “The Complexity Of Optimal Queuing Network Control,” *Math. Oper. Res*, vol. 24, pp. 293–305, 1999.

[15] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Prentice Hall, 1986.

[16] Y. Gai, B. Krishnamachari, and R. Jain, “Learning Multiuser Channel Allocations in Cognitive Radio Networks: A Combinatorial Multi-Armed Bandit Formulation,” in *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, pp. 1–9, April 2010.

Appendix A
Whittle Index for Two-State Markov Chains [3]

The steady state distribution of a continually passive load k from time t is given by $\bar{\pi}_k = \lim_{n\to\infty} \phi_k^n \pi_k(t) = \frac{Q_{01}^k}{1-(Q_{11}^k - Q_{01}^k)}$, where ϕ_k^n is the n-fold composition of ϕ_k. Assume that

1) $P_{01}^k \leq Q_{01}^k$, an unavailable load does not become available more likely after deployment.
2) $P_{01}^k \leq P_{11}^k$, an available active load is more likely to remain available than an unavailable active one to become so.
3) $Q_{01}^k \leq Q_{11}^k$, similar interpretation as above for passive loads.
4) $P_{11}^k \leq \bar{\pi}_k$, a continually passive load has a greater chance to be available than an available active load has of becoming so.

The Whittle index $w_k(\pi_k)$ for each load, provided its current belief state π_k, is then computed using its own parameters, where we have omitted the superscript of the load index for the simplicity of notation:

$$w(\pi) = \begin{cases} cA/B, & \text{if } \pi < \bar{\pi} \\ c\pi, & \text{if } \pi \geq \bar{\pi} \end{cases}$$

where $A = (\pi - \alpha\phi\pi)(1 - \alpha^{\tau_1+1}) + (1-\alpha)\alpha^{\tau 1+1}\phi^{\tau_1}P_{01}$ and $B = (\pi - \alpha\phi\pi)(\alpha^{\tau_2+1} - \alpha^{\tau_1+1}) + (1-\alpha)(1 + \alpha^{\tau 1+1}\phi^{\tau_1}P_{01} - \alpha^{\tau 2+1}\phi^{\tau_2}P_{11})$, and τ_1 and τ_2 are defined as $\tau_1 = \max\left\{\left\lceil \log_\Delta \frac{Q_{01}-\pi(1-\Delta)}{Q_{01}-P_{01}(1-\Delta)}\right\rceil, 0\right\}$ and $\tau_2 = \max\left\{\left\lceil \log_\Delta \frac{Q_{01}-\pi(1-\Delta)}{Q_{01}-P_{11}(1-\Delta)}\right\rceil, 0\right\}$ when $\Delta = Q_{11} - Q_{01} > 0$. In the case when $\Delta = 0$, τ_1 and τ_2 are given by

$$\tau_1 = \begin{cases} 0, & \text{if } \pi \leq P_{01} \\ 1, & \text{if } \pi > P_{01} \end{cases} \text{ and } \tau_2 = \begin{cases} 0, & \text{if } \pi \leq P_{11} \\ 1, & \text{if } \pi > P_{11} \end{cases}.$$

Appendix B
Approximations in Bayesian Inference from Aggregate Measurements [3]

Let $c = \sum_k c_k/N$, and then

$$\begin{aligned} p_k(1|y) &\approx \frac{\sum_{x^u:x_k=1} f(y; c\sum_j u_j x_j)p(x^u)}{p(y)} \\ &\approx \frac{\sum_{x^u:x_k=1} f(y; c\sum_j u_j x_j)p(x^u)}{\sum_{x^u} f(y; c\sum_j u_j x_j)p(x^u)} \\ &= \frac{\sum_{m=1}^K f(y; cm)p(\sum_{j:u_j=1} x_j = m, x_k = 1)}{\sum_{m=1}^K f(y; cm)p(\sum_{j:u_j=1} x_j = m)}. \end{aligned}$$

Assuming the independence among the states of loads, $p(x^u)$ is then given by a Poisson-Binomial distribution. Further applying the Poisson approximation, it follows that

$$p\Bigg(\sum_{j:u_j=1} x_j = m\Bigg) \approx \frac{\lambda_u^m e^{-\lambda_u}}{m!},$$

$$p\Bigg(\sum_{j:u_j=1} x_j = m, x_k = 1\Bigg) \approx \pi_k \frac{(\lambda_u - \pi_k)^{m-1} e^{-\lambda_u + \pi_k}}{(m-1)!}$$

where $\lambda_u = \sum_{k:u_k=1} \pi_k$. Moreover, when K is large, the poisson distribution can be approximated by the normal distribution with mean and variance λ^u (or $\lambda^u - \pi_k$) for computational efficiency in the above evaluation of p_k, which we use in our implementation.

Appendix C
The UCB1 Algorithm from [8], [16]

The UCB1 algorithm maintains two set of variables $\bar{x}_k$ and $\tau_k(t)$ over time, where $\bar{x}_k(t)$ is the empirical probability that load k is available when deployed and observed, and τ_k is the total number of times that load k has been dispatched over time t. UCB1 associates with each load an index given by the upper confidence bound $\bar{x}_k + \sqrt{\frac{L \ln t}{\tau_k}}$, hence the name, and the algorithm is detailed as follows.

```
UCB1
Parameter: A constant $L > 0$.
Initialization: Set $\bar{x}_k = 1/2$ and $\tau_k = 1$ for all $k \in [N]$
For time $t = 1, 2, \ldots, T$ do:
    1) Dispatch the $K$ loads with largest values of $\bar{x}_k + \sqrt{\frac{L \ln t}{\tau_k}}$.
    2) Observe the states $x_k$ of active loads. Update $\bar{x}_k$ by $\frac{\bar{x}_k \tau_k + x_k}{\tau_k + 1}$ and $\tau_k$ by $\tau_k + 1$ for each active load.
```