Online Learning For Demand Response

Dileep Kalathil and Ram Rajagopal

Abstract—**Demand response is a key component of existing and future grid systems facing increased variability and peak demands. Scaling demand response requires efficiently predicting individual responses for large numbers of consumers while selecting the right ones to signal. This paper proposes a new online learning problem that captures consumer diversity, messaging fatigue and response prediction. We use the framework of multi-armed bandits model to address this problem. This yields simple and easy to implement index based learning algorithms with provable performance guarantees.**

I. Introduction

Demand response is an important method for increasing the efficiency and reliability of power systems operations [1]. One main challenge for the effective implementation of a demand response program is the prediction and learning of consumers' behavior towards demand response signals. Response can depend on many consumer specific parameters like availability, temperature sensitivity, fatigue due to repeated scheduling and other stochastic factors like weather. Consumer specific parameters are typically private information. However, learning such parameters are essential for efficient scheduling of consumers in order to achieve a reliable load reduction via demand response programs.

In this paper, we address the problem of scheduling consumers for demand response under incomplete information about the expected load reductions from these consumers. We address this problem from the perspective of a demand response aggregator who manages a large number of consumers for demand response. The aggregator coordinates the demand response operations by sending a load curtailment signal to a set of selected consumers. Any consumer that fails to respond represents an additional cost for the aggregator. So, correctly predicting the load reduction for each consumer is very important to the aggregator.

When a consumer accepts the load curtailment signal from the aggregator, the resulting load reduction can be modeled as a random variable, a mean load reduction perturbed by a random noise. The mean load reductions of the consumer is unknown to the aggregator and has to be learned from the observed reductions. However, repeated scheduling can affect the response rate of the consumer. We model this fatigue effect as a multiplicative discount factor which depends on the number of times the consumer has been selected. The objective of the aggregator is to design a learning algorithm that will schedule the consumers optimally in such a way to maximize its total expected savings.

We address this learning and scheduling problem using the framework of non-Bayesian multi-armed bandits [2] [3]. Each consumer is treated as an arm in the sense of the classical multi-armed bandit problem. However, the (expected) reward from each arm depends on the number times that arm has been selected. In the classical multi-armed bandits problem, the reward from each arm is modeled as an i.i.d. sequence. Hence the optimal policy with complete information is to select the arm with the highest mean reward. The learning algorithms [2] [3] are in a way trying to approximate this optimal policy by selecting the arm with highest mean reward as much as possible. However in our case, due to the fatigue effect factor, selecting the arm with the highest mean reward all the time is no longer the optimal policy even with complete information. We show that in some cases selecting arms proportional to their mean is the optimal policy under complete information. We also propose a learning algorithm for proportional sharing when the mean is unknown.

Related work: In [2] Lai and Robbins introduced the classical non-Bayesian multi-armed bandit model where the reward from each arm is modeled as random variable with an unknown mean. They proposed an online learning algorithm and characterized the performance in terms of regret or the cost of learning. They also established a bound on the achievable performance and showed that their algorithm achieves that bound. Auer, et al. [3] proposed the celebrated UCB algorithm under the assumption that rewards are bounded. This algorithm has a simpler sample mean based index form and is easy to implement. Selecting multiple arms at the same time was addressed by Anantharam, et al. [4]. The problem of uniform estimation of the the mean values is addressed in the context of active learning by [5] and [6].

Anantharam, et al. [7] proposed a policy to the case where the arms are modeled as Markovian, not i.i.d. However, the state of the arm evolves only when the arm is selected. It remains frozen otherwise. These class of problems are called rested Markovian bandits problems. Tekin and Liu [8] showed that UCB policy can be extended to rested bandit problems. The class of problems when the state of arm continues to evolve even if it is not selected are called restless bandit problems. These problems are known to be PSPACE-hard [9]. Even in the case of complete information, i.e., when the transition kernel corresponding to each arm is known, an index based optimal policy may not exists [10]. In problems with some special structures, an index based optimal policy can be established [11]. Restless bandits problem which admits an index based optimal policy are called indexable.

Using bandit algorithms for learning consumers' behavior for demand response is relatively new. Taylor and Mathieu used a restless bandit framework [12] to model the load reduction from consumers. They showed that the problem is

Dileep Kalathil is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley. Ram Rajagopal is with the Department of Civil and Environmental Engineering, Stanford University.Emails: dileep.kalathil@berekeley.edu, ramr@stanford.edu.

indexable and computed the optimal scheduling policy. Jia et al. [13] used the linear bandits model to learn the consumers' unknown demand model as a response to dynamic pricing. They proposed a learning algorithm called piecewise linear stochastic approximation and characterized its performance. Our approach is different from these works.

II. Preliminaries and Problem Formulation

We consider the problem of sequential learning and scheduling of consumers for demand response (DR), from the perspective of a DR aggregator. We assume that the set of consumers is fixed during the scheduling time and is denoted by $\mathcal{N} = \{1, \ldots, i, \ldots, N\}$. If consumer i accepts the load curtailment signal from the aggregator, the load reduction from that consumer is modeled as

$$d_{i,t} = (\mu_i + w_{i,t}) \tag{1}$$

Here μ_i is the mean load reduction from consumer i if she accepts the DR signal. $w_{i,t}$ is a random noise to model the uncertainty in the load reduction. We assume that $\mu_i > 0$ and bounded, $w_{i,t}$ is zero mean and independent for all i and t. We also assume that $d_{i,t} \geq 0$ and is bounded for all i and t.

However, repeated scheduling can affect the load reduction from the consumers due to fatigue effects. And this will affect the net benefit of the aggregator. We model the effect of repeated scheduling by adding a user dependent discount factor to the reduction $d_{i,t}$. More precisely, we model the net benefit to the DR aggregator for selecting consumer i at time t as

$$r_{i,t} = f_i(n_{i,t})(\mu_i + w_{i,t}) \tag{2}$$

where $n_{i,t}$ is the number of times that the consumer i has been selected till time t and $f_i(\cdot)$ is a decreasing function. We assume that the DR aggregator knows $f_i(\cdot)$ for all $i \in \mathcal{N}$.

The discount factor $f_i(n_{i,t})$ can be thought as the actual decrease in the load reduction from the consumers due to the fatigue effect. For example, the probability that the consumer accepts a DR signal may decrease as she gets scheduled more and more. So, from the perspective of the aggregator, the expected value of savings (conditioned on the noise) will decrease w.r.t. $n_{i,t}$.

At each time t, the DR aggregator can select K consumers out of the total N consumers. This is specified by a control vector $a_t = (a_{i,t}, 1 \leq i \leq N)$. $a_{i,t} = 1$ if a consumer i selected at time t and zero otherwise. The DR aggregator has no information about the mean load reduction $\mu_i, i \in \mathcal{N}$. So, it has to learn μ_is from the observed reductions. The reduction $d_{i,t}$ from a consumer i is observed only if that consumer is selected at time t, i.e., $a_{i,t} = 1$.

The objective of the aggregator is to solve the following sequential optimization problem

$$\max_{a_t, 1 \leq t \leq T} \quad \mathbb{E}\left[\sum_{i=1}^{N}\sum_{t=1}^{T} f_i(n_{i,t}) d_{i,t} a_{i,t}\right] \tag{3}$$

$$\text{s.t.} \quad \sum_{\tau=1}^{(t-1)} a_{i,\tau} = n_{i,t}, \forall i, \forall t \tag{4}$$

$$\sum_{i=1}^{N} a_{i,t} = K, \forall i, \forall t \tag{5}$$

In the following, we assume that the form of the discounting function is same for all users, i.e., $f_i(\cdot) = f(\cdot)$ for all i. This assumption is mainly for the notational convenience and clarity of presentation.

III. Scheduling with Complete information

The problem specified in (3)-(5) is a sequential stochastic optimization and typically difficult to solve. In order to get an intuition about the optimal scheduling polices for the above problem, we first solve a deterministic version of this problem.

In this section, we assume that DR aggregator has complete information and no uncertainty in the system. More precisely, we assume that the aggregator knows $\mu_i, \forall i$. Also, $w_{i,t} = 0, \forall i, \forall t$. We first solve the case with $K = 1$, i.e., selecting only one consumer at a time. The case where $K > 1$ is described in section III-C.

From the perspective of the aggregator, the (discounted) load reduction is equivalent to a reward it is getting. So, in the following, we will use the terms reward and reduction interchangeably. This is mainly to use the same terminologies that are standard in the learning and scheduling literature.

Let

$$t_i = \sum_{t=1}^{T} a_{i,t} \tag{6}$$

So, t_i is the number of times consumer i is selected till time T. When there is no uncertainty, $d_{i,t} = \mu_i$ for all i, t. So, in the deterministic setting with complete information and with $K = 1$, we can rewrite the the optimization problem (3)-(5) as,

$$\max_{t_i, 1 \leq i \leq N} \quad \sum_{i=1}^{N}\sum_{\tau=1}^{t_i} f(\tau)\mu_i \tag{7}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} t_i = T \tag{8}$$

This an integer optimization problem and it is difficult to characterize its solution analytically. So, we consider a relaxed version of the above problem. We write,

$$\max_{t_i, 1 \leq i \leq N} \quad \sum_{i=1}^{N}\int_{\tau=1}^{t_i} f(\tau)\mu_i \tag{9}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} t_i = T \tag{10}$$

The optimal solution is denoted as $t_i^*, 1 \leq i \leq N$.

We will consider the solution of the above relaxed problem as the benchmark for quantifying the performance of our learning algorithm in Section IV.

We consider two different discount functions. First we consider a slowly decreasing function, $f(\tau) = 1/\tau$. Then we consider an exponentially deceasing function, $f(\tau) = \beta^{\tau}$ for some $0 < \beta < 1$.

A. *Inverse Linear Discount:* $f(\tau) = 1/\tau$

We show that when the discount factor is inversely proportional to the number of times the consumer has been selected, then the optimal scheduling policy is *proportional sharing*. Each consumer i is selected in proportion to its mean reduction μ_i. We make this precise in the following proposition.

Proposition 1. *The optimal solution of the problem* (9)-(10) *with discount function* $f(\tau) = 1/\tau$ *is*

$$t_i^* = \frac{\mu_i T}{\mu_{sum}} \tag{11}$$

where $\mu_{sum} = \sum_{j=1}^{N} \mu_j$.

Proof. Lagrangian of (9)-(10) is

$$L(\lambda, t_i, 1 \leq i \leq N) = \sum_{i=1}^{N} \mu_i \log t_i + \lambda \left(\sum_{i=1}^{N} t_i - T \right).$$

Using KKT conditions, we get $(\mu_i / t_i^*) + \lambda^* = 0, \forall i$. Taking summation w.r.t. i, $\lambda^* = -(\mu_{sum}/T)$. Substituting back, we get, $t_i^* = (\mu_i T / \mu_{sum})$. □

Even though the optimal selection of each consumer, t_i^*, is given by the above optimization, it is not immediately clear which is a good algorithm that achieves proportional sharing for any given time t. Below, we give a simple online algorithm (Algorithm 1) which achieves proportional sharing allocation for any given t, upto rounding-off errors. We also show that this is the optimal online algorithm for solving the optimization problem (7)-(8) for any arbitrary $f(\cdot)$.

Algorithm 1 Greedy Algorithm for Proportional Selection (Deterministic)

```
Input: Mean rewards $\mu_i, 1 \leq i \leq N$.
  1) Assign $t \leftarrow 0, n_{i,t} \leftarrow 1, \forall i$.
     Assign index $g_{i,t} = f(n_{i,t})\mu_i, \forall i$.
  2) Select consumer $i^* = \arg\min_{i, 1 \leq i \leq N} g_i$
     Get reward $r_{i^*,t} = f(n_{i^*,t})\mu_{i^*}$
  3) Update $t \leftarrow t + 1$, $n_{i^*,t} \leftarrow n_{i^*,t} + 1$. Update $g_{i^*,t}$.
  4) IF $t < T$, go to step 2. ELSE, end.
```

We first give an illustration of the algorithm in Table I with $f(\tau) = 1/\tau$. Suppose $N = 3$ with $\mu_1 = 8, \mu_2 = 4, \mu_3 = 1$. The first column is the time index, g_i is the index of consumer i and the last column indicates the selected consumer in each round. Ties are broken towards the most selected consumer. This is a greedy algorithm because in each round, the algorithm selects the consumer with the highest index.

Note that when the consumer 3 is selected for the first time (when $t = 7$), consumer 1 and consumer 2 have already been selected 4 times and 2 times respectively. Similarly, when consumer 2 is selected the first time (when $t = 3$), consumer 1 has already been selected 2 times. So, at any t, consumers are selected in proportion of their mean μ_i upto a rounding-off error. It is not difficult to see that this selection procedure operates in cycles and the same proportion is maintained.

We now make this precise in the following proposition.

TABLE I
EXAMPLE OF ALGORITHM 1

Time	g_1	g_2	g_3	Selected
t=1	8	4	2	C-1
t=2	8/2	4	2	C-1
t=3	8/3	4	2	C-2
t=4	8/3	4/2	2	C-1
t=5	8/4	4/2	2	C-1
t=6	8/5	4/2	2	C-2
t=7	8/5	4/3	2	C-3

Proposition 2. *(i) Algorithm 1 with* $f(\tau) = 1/\tau$ *achieves a proportional sharing allocation.*
(ii) Algorithm 1 is the optimal online algorithm for the optimization problem (7)-(8).

Proof. (sketch) (i) Without loss of generality, assume that $\mu_1 > \ldots > \mu_i > \ldots > \mu_N$. Note that if consumer N selected t_N times, consumer 1 should have been selected at least $\lceil (\mu_1 t_N / \mu_N) \rceil$ time but not more than $\lceil (\mu_1 t_N / \mu_N) \rceil + 1$. So, neglecting the rounding-off error, we get $(\mu_1 / t_1) = (\mu_N / t_N)$. This argument can be extended for all i, j and hence $(\mu_i / t_i) = (\mu_j / t_j) = \lambda$, where λ is some constant. From this, by using the same step as in the proof of Proposition 1, we get $t_i = (\mu_i T / \mu_{sum})$.
(ii) Note that when consumer i is selected at time t, then the aggregator gets a reward $g_{i,t} = f(n_{i,t})\mu_i$. So, Algorithm 1 is a greedy algorithm.

Let $n_{i,\tau}$ and $\tilde{n}_{i,\tau}$ be the number of times consumer i selected till time τ by the greedy algorithm and the other algorithm respectively. Also, let $r_\tau, \tau \leq t$ and $\tilde{r}_\tau, \tau \leq t$ be the sequence of rewards obtained by the greedy algorithm and the other algorithm. Note that the rewards here are, $r_\tau = I\{i \text{ is selected}\} f(n_{i,\tau})\mu_i$. The rewards $\tilde{r}_\tau$ are defined similarly. Note that the reward from each consumer is a deterministic sequence with decreasing values. So, it is not difficult to observe that if $\tilde{r}_t \geq r_t$ for some t, then there exists a time index τ such that $\tau < t$ and $r_\tau = \tilde{r}_t$.

Let J_t be the cumulative reward accrued by the greedy algorithm and let $\tilde{J}_t$ be be the cumulative reward accrued by any other algorithm. Then using the above argument, $J_t \geq \tilde{J}_t$ for all t. So, the greedy algorithm is optimal. □

B. *Exponential Discount:* $f(\tau) = \beta^{\tau}$

We consider another discount function, $f(\tau) = \beta^{\tau}$, which decays to zero at an exponential rate. We show that the rate of decay has a significant effect on scheduling. First we characterize the optimal solution (9)-(10) with this exponential discount function.

Proposition 3. *The optimal solution of the problem* (9)-(10) *with discount function $f(\tau) = \beta^{(\tau)}$ is*

$$t_i^* = \frac{T}{N} + \frac{1}{\log(1/\beta)} \log \frac{\mu_i}{(\mu_{prod})^{1/N}} \tag{12}$$

Proof. Writing Lagrangian and using KKT conditions with $f(\tau) = \beta^{(\tau)}$, we get $\mu_i \beta^{t_i^*} = -\lambda^*$. Taking product w.r.t. the index i and rearranging, we get $-\lambda^* = (\mu_{prod})^{1/N} \beta^{T/N}$. Substituting this in the KKT condition and solving, we get,

$$\mu_i \beta^{t_i^*} = (\mu_{prod})^{1/N} \beta^{T/N}, \forall i. \tag{13}$$

The result follows. □

Remark 1. *When the discount function is $1/\tau$, the ratio t_i^*/t_j^* is μ_i/μ_j for any arbitrary $i, j \in \mathcal{N}$ (c.f.* (11)*). With an exponential discount function β^τ, this ratio approaches 1 for large T, i.e. $(t_i^*/t_j^*) \to 1$ as $T \to \infty$ (c.f.* (12)*).*

Using the part (ii) of Proposition 1, it can be shown that the greedy algorithm (Algorithm 1) is the optimal online algorithm to achieve the allocation as specified by (11).

C. Selecting Multiple Consumers: $K > 1$

When $K > 1$, it is not straight forward to write the optimization problem (3)-(5) in terms of t_is as in (7)-(8) and get a closed form expression. We will address this problem in a future work.

However, it is easy to see that a greedy policy is optimum for this problem. The argument is similar to part (ii) of Proposition 2 and hence we skip the proof. The greedy algorithm will be similar to Algorithm 1 for $K = 1$. At each time t, aggregator will have an index $g_{i,t}$ for each consumer. Now, instead of selecting only one consumer who has the highest index, here the aggregator will select K consumers with the highest indices. In section V we show the simulation results when $K > 1$.

IV. Learning Algorithm

In this section we consider the case where the aggregator doesn't know μ_is and the observations are corrupted by noise. So, the aggregator has to learn the μ_is from the noisy observations and use this for making the scheduling decision in the next time instant. The objective of the aggregator is characterized in the optimization problem (3)-(5). Here we propose an *online learning algorithm* to solve this optimization problem.

The performance of an online learning algorithm is typically quantified using a metric called regret. Regret is essentially a comparison between the performance of the optimal algorithm with complete information and the performance of the learning algorithm. Here we define two types of regret, share regret and cost regret.

Let t_i^* be the number of time consumer i is selected till time T by the optimal algorithm with complete information. This t_i^* is indeed the solution of the optimization problem (9)-(10) and we have characterized the value of t_i^* in Section III for $f(\tau) = 1/\tau$ and $f(\tau) = \beta^\tau$. Let t_i be the number of times consumer i is selected till time T by the learning algorithm. We define the share regret $R_S(T)$ as

$$R_S(T) = \mathbb{E}\left[\sum_{i=1}^{N} |t_i^* - t_i|\right] \tag{14}$$

Since we are mainly interested in the net expected savings of the aggregator, we define the cost regret to characterize this quantity. Let

$$J^* = \sum_{i=1}^{N} \int_{\tau=1}^{t_i^*} f(\tau)\mu_i, \quad J = \sum_{i=1}^{N} \int_{\tau=1}^{t_i} f(\tau)\mu_i \tag{15}$$

Note that, from (9)-(10), J^* and J are the expected savings for the aggregator under the optimal algorithm (Algorithm 1) and the learning algorithm respectively. We define the cost regret $R_C(T)$ as

$$R_C(T) = \mathbb{E}\left[|J - J^*|\right] \tag{16}$$

In section III we characterized a closed form expression for $K = 1$ and $f(\tau) = 1/\tau$. So, here also, we restrict our attention to the same case. In Section V we show the simulation results for the case $K > 1$.

When $f(\tau) = 1/\tau$, note that

$$J - J^* = \sum_{i=1}^{N} \sum_{i=1}^{M} \mu_i \log \frac{t_i}{t_i^*} \tag{17}$$

We propose an index based learning algorithm (Algorithm 2). It is similar to the deterministic algorithm (Algorithm 1) except in the form of the index. In Algorithm 1, index $g_{i,t}$ is given by $(1/n_{i,t})\mu_i$. But, here μ_i is unknown. So, we use the index

$$g_{i,t} = \frac{1}{n_{i,t}} \left(\hat{\mu}_{i,t} + \sqrt{\frac{M \log T}{(n_{i,t} + 1)}} \right). \tag{18}$$

Here $\hat{\mu}_{i,t}$ sample average of the reduction form the consumer i till time t, i.e., $\hat{\mu}_{i,t} = \left(\sum_{\tau=1}^{t} d_{i,t}\right)/(n_{i,t})$. Thus, $\hat{\mu}_i$ is an unbiased estimate of μ_i. The second term is called the 'upper confidence term'. This form of index, sum of an unbiased estimate and an upper confidence term, is standard in the multi-armed bandit literature.

Algorithm 2 Proportional Learning Algorithm

```
1) Select each consumer once. Select M ¿ 2.
   Update $t \leftarrow N$, $n_{i,t} \leftarrow 1, \forall i$.
   Compute the index $g_{i,t}, \forall i$.
2) Select the consumer $i^* = \arg\max_{i, 1 \le i \le N} g_{i,t}$.
3) Update $t \leftarrow t + 1$, $n_{i^*,t} \leftarrow n_{i^*,t} + 1$, Update $g_{i^*,t}$.
4) IF $t < T$, go to step 2. ELSE, end.
```

Theorem 1. *Let $R_S(T)$ and $R_C(T)$ be as defined in* (14) *and* (16) *respectively. Then, with $M > 2$, Algorithm 2 achieves*

$$R_S(T) \le C_1 \frac{N}{(\mu_{min})^{1/2}} \sqrt{\frac{4MT \log T}{\mu_{sum}}} \tag{19}$$

$$R_C(T) \le C_2 \frac{N}{(\mu_{min})^{1/2}} \sqrt{\frac{4M \log T}{T}} \tag{20}$$

where $\mu_{min} = \min_i \mu_i$ and C_1 and C_2 are constants which are independent of N and T.

Proof. Define the events

$$\mathcal{E}_{i,t} = \left\{ \omega : |\hat{\mu}_{i,t} - \mu_i| \le \sqrt{\frac{M \log T}{(n_{i,t}+1)}} \right\} \tag{21}$$

$$\mathcal{E} = \cap_{i=1}^{N} \cap_{t=1}^{T} \mathcal{E}_{i,t} \tag{22}$$

In the following, we will restrict ourselves to the set $\mathcal{E}$.

Using Chernoff-Hoeffding inequality,

$$\mathbb{P}(\mathcal{E}_{i,t}^c) \le 2e^{-2M \log T (n_{i,t}/(n_{i,t}+1))} \le 2e^{-M \log T}, \forall i, \forall t.$$

Let $\delta := 2e^{-M \log T}$. Then,

$$\mathbb{P}\left(\mathcal{E}\right) = 1 - \mathbb{P}\left(\mathcal{E}^c\right) \ge (1 - NT\delta) \tag{23}$$

Consider an arbitrary consumer j and let t be the time when consumer j is selected for the final time. So, $n_{j,t} = t_j - 1$. Then, on $\mathcal{E}$,

$$\begin{aligned} g_{j,t} &= \frac{1}{(n_{j,t}+1)} \left(\hat{\mu}_{j,t} + \sqrt{\frac{M \log T}{(n_{j,t}+1)}} \right) \\ &= \frac{1}{t_j} \left(\hat{\mu}_{j,t} + \sqrt{\frac{M \log T}{t_j}} \right) \\ &\le \frac{1}{t_j} \left(\mu_j + 2\sqrt{\frac{M \log T}{t_j}} \right). \end{aligned} \tag{24}$$

We get the last inequality because (21) and since we are considering the events only in $\mathcal{E}$.

Also, for any arbitrary consumer i,

$$\begin{aligned} g_{i,t} &= \frac{1}{(n_{i,t}+1)} \left(\hat{\mu}_{i,t} + \sqrt{\frac{M \log T}{(n_{i,t}+1)}} \right) \\ &\ge \frac{\mu_i}{(n_{i,t}+1)} \ge \frac{\mu_i}{t_i} \end{aligned} \tag{25}$$

The first inequality is also due to (21).

Since consumer j is selected at time t, its index should be the highest at that time, i.e., $g_{j,t} \ge g_{i,t}, \forall i$. So, using (24) and (25),

$$\frac{\mu_i}{t_i} \le \frac{\mu_j}{t_j} + \frac{2}{t_j}\sqrt{\frac{M \log T}{t_j}}. \tag{26}$$

Note that the above inequality is independent of the time index t.

Multiply both sides of (26) by t_i and take summation w.r.t. the index i. After some manipulations we get,

$$\frac{\mu_{sum}}{T} \le \frac{\mu_j}{t_j} + \frac{2}{t_j}\sqrt{\frac{M \log T}{t_j}} \tag{27}$$

and rearranging,

$$t_j \le \frac{\mu_j}{\mu_{sum}} T + \frac{2T}{\mu_{sum}}\sqrt{\frac{M \log T}{t_j}}. \tag{28}$$

Since $t_j^* = (\mu_j/\mu_{sum})T$, using this in the above inequality, we get

$$t_j - t_j^* \le \frac{2T}{\mu_{sum}}\sqrt{\frac{M \log T}{t_j}} \tag{29}$$

Suppose that consumer j selected more than its 'fair-share', i.e., $t_j \ge t_j^* = (\mu_j T)/\mu_{sum}$ (otherwise, the above inequality is trivial). Using this in the right hand side of the above inequality,

$$t_j - t_j^* \le \sqrt{\frac{4M(T \log T)}{\mu_j \mu_{sum}}} \tag{30}$$

Now, consider another consumer i who is selected less than its 'fair-share', i.e., $t_i \le t_i^* = (\mu_i T)/\mu_{sum}$. There there exists another consumer j who is selected more than its 'fair-share', i.e., $t_j \ge t_j^* = (\mu_j T)/\mu_{sum}$. Since inequality (26) is for any arbitrary i, j, use the fact that $t_j \ge t_j^* = (\mu_j T)/\mu_{sum}$ on the right hand side of (26) we get

$$\frac{\mu_i}{t_i} \le \frac{\mu_{sum}}{T} + \sqrt{\frac{4M(\mu_{sum})^3 \log T}{(\mu_j)^3 T^3}} \tag{31}$$

Multiplying both sides by (T/μ_{sum}) and using the fact that $t_i^* = (\mu_i T)/\mu_{sum}$, we get

$$\frac{t_i^*}{t_i} \le 1 + \sqrt{\frac{4M(\mu_{sum}) \log T}{(\mu_j)^3 T}} \tag{32}$$

and using the fact that $t_i \le t_i^* = (\mu_i T)/\mu_{sum}$, this implies

$$t_i^* - t_i \le \sqrt{\frac{4M(T \log T)}{\mu_j \mu_{sum}}} \tag{33}$$

Combining (30) nd (33), for any arbitrary consumer j, on the event $\mathcal{E}$,

$$|t_j - t_j^*| \le \sqrt{\frac{4M(T \log T)}{\mu_j \mu_{sum}}} \tag{34}$$

Now,

$$\begin{aligned} \mathbb{E}\left[\sum_{i=1}^{N} |t_i - t_i^*|\right] &\le \frac{N}{(\mu_{min})^{1/2}}\sqrt{\frac{4MT \log T}{\mu_{sum}}}\mathbb{P}(\mathcal{E}) \\ &\quad + T\ \mathbb{P}(\mathcal{E}^c) \\ &\le \frac{N}{(\mu_{min})^{1/2}}\sqrt{\frac{4MT \log T}{\mu_{sum}}} + \frac{N}{T^{M-2}} \end{aligned} \tag{35}$$

This is the result in (19).

In order to get (20), we do the following steps. For an 'over-selected' consumer j, from (30)

$$\frac{t_j}{t_j^*} \le 1 + \frac{1}{t_j^*}\sqrt{\frac{4M(T \log T)}{\mu_j \mu_{sum}}} = 1 + \sqrt{\frac{4M\mu_{sum}(\log T)}{(\mu_j)^3 T}}$$

which gives

$$\mu_j \log \frac{t_j}{t_j^*} \le \sqrt{\frac{4M\mu_{sum}(\log T)}{(\mu_j) T}} \tag{36}$$

For an 'under-selected' consumer, from (32),

$$\mu_i \log \frac{t_i^*}{t_i} \le \sqrt{\frac{4M\mu_{sum}(\log T)}{(\mu_i) T}} \tag{37}$$

Combining (36) and (37), on the event $\mathcal{E}$ and upperbounding as in (35) we get (20).

□

V. Simulations

In this section we show the performance of our learning algorithm using numerical simulations. We consider a setting with the 1000 consumers, $N = 1000$. We assume that the mean reductions μ_is are in the interval $[0.1, 5.1]$ and without loss of generality, consumers are ordered in the increasing order of μ_i. The noise $w_{i,t}$ are generated according to a uniform distribution.

Figure 1 shows the share regret, $R_S(T)$, as compared to the upper bound in equation (19). The y axis is $\|t_i^* - t_i\|_1 = \sum_i^N |t_i^* - t_i|$ and the x axis is the time index. The blue line shows the performance of our learning algorithm, which is averaged over 40 simulation runs. The share regret increases with the time.

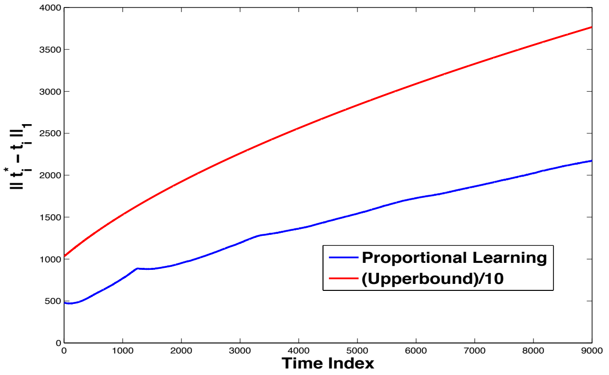

Fig. 1. Share-Regret

Figure 2 shows the the cost regret, $R_C(T)$, as compared to the upper bound in equation (20). The blue line shows the performance of our learning algorithm, the value of $|J - J^*|$ averaged over 40 simulation runs. We also marks the error bar which shows the 95% confidence interval. The simulation suggests that the cost regret indeed decreases for large time index, as suggested by our result.

An intuitive explanation of this is as follows. From (17), note that $J - J^*$ is a sum of the terms of the form $\log \frac{t_i}{t_i^*}$. From the first part of Theorem 1, we have $|t_i - t_i^*|$ is $O(\sqrt{T} \log T)$. Since t_i^* is $\mu_i T / \mu_{sum}$, we know that t_i^* is $O(T)$. So, we get, $\log \frac{t_i}{t_i^*} \leq \log(1 + \frac{|t_i - t_i^*|}{t_i^*})$, which is $O\left(\log(1 + \sqrt{\log T / T})\right)$ from the above argument. This goes to zero as T goes to infinity.

Note that the time index is on x axis is large for both figures as an artifact of the fact that we consider the case $K = 1$. In fact, the actual number of selection for each consumer is low. This is shown in Figure 3. x axis shows the consumer number and y axis is the number of times each consumer is selected. The blue line is the number of selection according to our learning algorithm and the red line is the optimal (proportional) selection. Note that the maximum number of selection is less than 20.

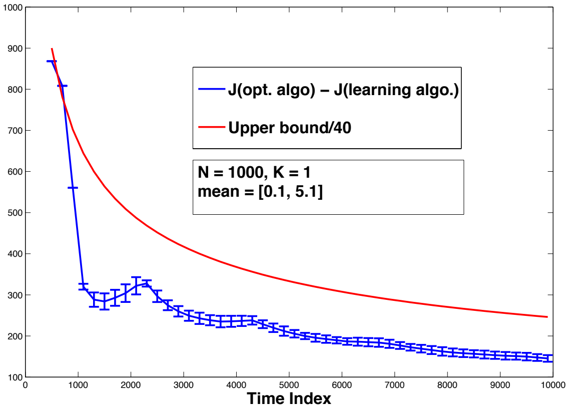

Fig. 2. Cost-Regret

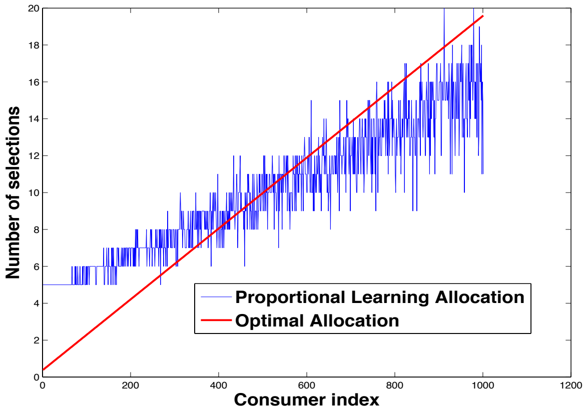

Fig. 3. Number of Selections

Figure 4 shows a simulation with $K = 50$. We plot the efficiency of our learning algorithm, J/J^* as a function of time index. Note that, as expected, the efficiency increases with time. This suggests that our algorithm works even for the case $K > 1$.

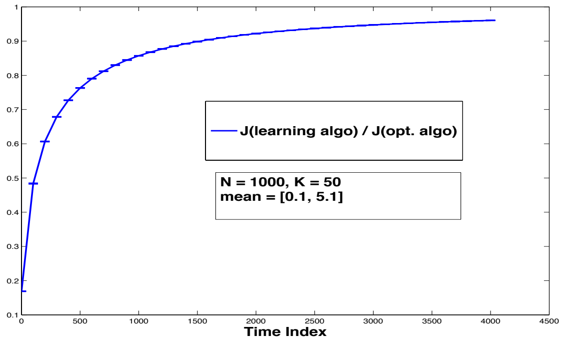

Fig. 4. Efficiency

VI. Conclusion

We addressed the problem of learning consumers' behavior from observed load reductions using the framework of multi-armed bandits. We showed that, in some cases, the optimal scheduling policy is proportional sharing and proposed an online learning algorithm for this.

We consider this only as a starting pointing for designing practical and efficient algorithm for learning consumers' behavior for demand response programs. In the future works, we will address the problem of online clustering of consumers according their responses and use this clustering to improve the prediction. We will also include the effect of the real time prices in the scheduling decision. Using the historical demand response data from DR aggregators to validate and improve our learning algorithms is another immediate objective.

References

[1] P. Siano, "Demand response and smart grids - a survey," *Renewable and Sustainable Energy Reviews*, vol. 30, pp. 461–478, 2014.

[2] T. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4-22, 1985.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2, pp. 235-256, 2002.

[4] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically efficient allocation rules for the multi-armed bandit problem with multiple plays - part i: i.i.d. rewards," *IEEE Transactions on Automatic Control*, vol. 32, no. 11, pp. 968-975, November, 1987.

[5] A. Antos, V. Grover, and C. Szepesvári, "Active learning in heteroscedastic noise," *Theoretical Computer Science*, vol. 411, no. 29, pp. 2712–2728, 2010.

[6] A. Carpentier, A. Lazaric, M. Ghavamzadeh, R. Munos, and P. Auer, "Upper-confidence-bound algorithms for active learning in multi-armed bandits," in *Algorithmic Learning Theory*. Springer, 2011, pp. 189–203.

[7] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically efficient allocation rules for the multi-armed bandit problem with multiple plays - part ii: Markovian rewards," *IEEE Transactions on Automatic Control*, vol. 32, no. 11, pp. 977-982, November 1987.

[8] C. Tekin and M. Liu, "Online learning of rested and restless bandits," *IEEE Transactions on Information Theory*, vol. 58, no. 8, pp. 5588–5611, 2012.

[9] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of optimal queuing network control," *Mathematics of Operations Research*, vol. 24, no. 2, pp. 293–305, 1999.

[10] P. Whittle, "Restless bandits: Activity allocation in a changing world," *Journal of applied probability*, pp. 287–298, 1988.

[11] K. Liu and Q. Zhao, "Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5547–5567, 2010.

[12] J. Taylor and J. Mathieu, "Index policies for demand response," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1287–1295, 2014.

[13] L. Jia, L. Tong, and Q. Zhao, "An online learning approach to dynamic pricing for demand response," *arXiv preprint arXiv:1404.1325*, 2014.

[14] D. Kalathil and R. Rajagopal, "Online learning for demand response," *available at https://bcci.me.berkeley.edu/∼kalathil/publications.html*, October, 2015.