2018 IEEE Conference on Decision and Control (CDC)
Miami Beach, FL, USA, Dec. 17-19, 2018

Learning and Selecting the Right Customers for Reliability: A Multi-armed Bandit Approach

Yingying Li, Qinran Hu, and Na Li

Abstract— **In this paper, we consider residential demand response (DR) programs where an aggregator calls upon some residential customers to change their demand so that the total load adjustment is as close to a target value as possible. Major challenges lie in the uncertainty and randomness of the customer behaviors in response to DR signals, and the limited knowledge available to the aggregator of the customers. To learn and select the right customers, we formulate the DR problem as a combinatorial multi-armed bandit (CMAB) problem with a reliability goal. We propose a learning algorithm: CUCB-Avg (Combinatorial Upper Confidence Bound-Average), which utilizes both upper confidence bounds and sample averages to balance the tradeoff between exploration (learning) and exploitation (selecting). We prove that CUCB-Avg achieves $O(\log T)$ regret given a time-invariant target. Simulation results demonstrate that our CUCB-Avg performs significantly better than the classic algorithm CUCB (Combinatorial Upper Confidence Bound).**

I. Introduction

Demand response (DR) has been playing an increasing role in reducing the operation cost and improving the sustainability of the power grid [1]–[9]. Most of the existing successful DR programs are for commercial and industrial customers. As residential demand takes up to almost 40% of the U.S. electricity consumption [10], there is a growing effort in designing residential DR in both academia and industry. In a typical residential DR program, there is a DR aggregator such as a utility company requests load changes from users, for example, by changing the temperature set points of the air conditioners. To encourage users' participation, most of the residential DR programs use incentive schemes such as prices, rewards, coupons, raffles, etc, under the assumption that customers are price responsive [7]–[9]. However, because the average monetary reward budget for single household is usually small, it is reported that rewards play a limited role for users to decide whether to participate in or opt out of a DR program [8].

On the other side, there are many factors besides rewards that affect residential user decisions, such as house size and type, household demographics, outdoor humidity and temperature, people's lifestyles, etc. However, the DR aggregator has limited knowledge of these factors. It is also unclear how these factors will affect people's DR action. Moreover, people with similar factors might react to the same DR signal in very different ways. These intrinsic, heterogeneous uncertainties associated with the residential customers call for learning approaches to understand and interact with the customers in a smarter way.

Multi-armed bandit (MAB) emerges as a natural framework to handle such uncertainties [11], [12]. In the simplest setting, MAB considers n independent arms, each providing a random contribution according to its own distribution at time $1 \leq t \leq T$. Without knowing these distributions, a decision maker picks one arm at each time step, and tries to maximize the total expected contribution. The decision maker should decide whether to *explore* arms to learn the unknown distribution, or to *exploit* the current knowledge by selecting the arm that has been providing the highest contribution. When the decision maker can select multiple arms at each time, the problem is referred as CMAB (Combinatorial MAB) in literature [13]–[17]. (C)MAB captures a fundamental tradeoff in most learning problems: *exploration vs. exploitation*. A common metric to evaluate the performance of (C)MAB learning algorithms is the regret, which captures the difference between the optimal value assuming the distributions are known and the achieved value of the online learning algorithm. A sublinear regret implies good performance because it indicates that the learning algorithm eventually learns the optimal solution.

When applying CMAB framework to residential demand response, we can treat each customer as one arm. Then the aggregator follows CMAB methods to explore (learn) and exploit (select) the customers to achieve the goal of its DR program. There exist studies of DR via (C)MAB [9], [18]–[20]. However, most literature sets the goal as maximizing the load reduction for peak hours without considering the load reduction target and reliability issues.

Our Contributions: In this paper, we formulate the DR as a CMAB problem whose objective is to minimize the deviation between the total load adjustment and a target level for the power system reliability. We consider a large number of residential customers, each of whom can commit one unit of load change (either reduction or increase) with an unknown probability. The task of the aggregator is to select a subset of the customers to approximate the target level as close as possible. The size of the subset is not fixed, giving flexibility to the aggregator to achieve different target levels. Compared with the classic CMAB literature [13]–[17], a major difference of our formulation is that the reliability objective leads to a non-monotonic objective function for the CMAB problem, making the existing CMAB approaches and regret analysis inapplicable here.

In order to design our CMAB online learning algorithm, we first study the corresponding offline combinatorial optimization problem assuming the probabilities of customers

The work was supported by NSF 1608509, NSF CAREER 1553407, AFOSR YIP, and ARPA-E through the NODES program. Y. Li, Q. Hu, and N. Li are with the School of Engineering and Applied Sciences, Harvard University, 33 Oxford Street, Cambridge, MA 02138, USA (email: yingyingli@g.harvard.edu, qinranhu@g.harvard.edu, nali@seas.harvard.edu).

are known. Based on the structure of the offline algorithm, we propose an online algorithm CUCB-Avg (Combinatorial Upper Confidence Bound-Average) and provide a rigorous regret analysis. We show that, over T time steps, CUCB-Avg achieves $O(\log T)$ regret given a static target. The dependence of regret on dimension n (the number of customers) is polynomial. By simulation, we show that the performance of CUCB-Avg is much better than the classic algorithm CUCB [13], [16], [17] and similar to Thompson sampling, another popular CMAB method which has good empirical performance but is usually difficult to theoretically analyze [21]–[23].

Related Work in CMAB. Most literature in CMAB studies a classic formulation which aims to maximize the total (weighted) contribution of K arms with a fixed integer K (and known weights) [12], [14]–[17], [24]. As for more general problem formulation, Chen et. al. study the monotone objective function: the objective value function is monotonically nondecreasing with arms' parameters given a fixed selected subset [13]. They propose the Combinatorial Upper Confidence Bound (CUCB) using the principle of *optimism in the face of uncertainty*. Another line of work follows the *Bayesian* approach. [25] studies a Bayesian learning algorithm, Thompson sampling, for general CMAB problems, but its analysis is based on several assumptions including finite prior distribution and the uniqueness of the optimal solution, and the regret bound consists of a large exponential term.

However, our CMAB problem with the reliability objective function does not satisfy the conditions of monotonicity or the uniqueness of the optimal solution, and a properly selected prior distribution for our problem may not satisfy the assumption in [25]. Therefore, either the learning approaches or the analysis in current literature do not suit our CMAB problem, motivating us to design new CMAB algorithms.

Organization of the Paper. Section II introduces the problem formulation. Section III introduces an offline algorithm and an online algorithm CUCB-Avg. Section IV studies the performance guarantee for a time-invariant target. Section V provides the simulation results.

Notations. Given a set E, and a universal set U, the complement of set E is denoted as $\bar{E}$, the cardinality of set E is $|E|$. For any positive integer n, let $[n] = \{1, \dots, n\}$. Let $I_E(x)$ denote the indicator function on set U such that $I_E(x) = 1$ if $x \in E$ and $I_E(x) = 0$ if $x \notin E$. When $k = 0$, the summation $\sum_{i=1}^{k} a_i = 0$ for any a_i, and the set $\{\sigma(1), \dots, \sigma(k)\} = \emptyset$ for any $\sigma(i)$. Finally, we define the big-O and small-o notations. For $x = (x_1, \dots, x_k) \in \mathbb{R}^k$, we write $f(x) = O(g(x))$ as $x \to +\infty$ if there exists a constant M such that $|f(x)| \leq M|g(x)|$ for any x such that $x_i \geq M \ \forall i \in [k]$; and we write $f(x) = o(g(x))$ if $\lim_{x \to +\infty} f(x)/g(x) = 0$. We usually omit the phase "as $x \to +\infty$" for simplicity. When studying the asymptotic behavior near zero, we consider the inverse of x.

II. Problem Formulation

Motivated by the discussion in the previous section, we will formulate the DR as a CMAB problem in this section. We focus on the load reduction to illustrate the problem. The load increase can be treated in the same way.

Consider a demand response (DR) program with an aggregator and n residential customers (arms) over T time steps where each time step corresponds to one DR event.[1] Each customer may respond to a DR event by reducing one unit of power consumption with probability $0 \leq p_i \leq 1$, or not respond at all. The demand reduction by customer i at time step t is denoted by $X_{t,i}$, which is assumed to follow Bernoulli distribution: $X_{t,i} \sim \text{Bern}(p_i)$ and is independent across time[2].

At each time $1 \leq t \leq T$, there is a DR event with a demand reduction target $D \geq 0$ determined by the power system. This reduction target might be due to a sudden drop of renewable energy generation or a peak load reduction request, etc. The aggregator aims to select a subset of customers $S_t \subseteq [n]$, such that the total demand reduction is as close to the target as possible. The loss/cost at time t can be captured by the squared deviation of the total reduction from the target D:

$$L_t(S_t) = (\sum_{i \in S_t} X_{t,i} - D)^2$$

Since demand reduction $X_{t,i}$ are random, the goal is to minimize the expected squared deviation,

$$\min_{S_t \subseteq [n]} \mathbb{E}\, L_t(S_t). \tag{1}$$

When the response probability profile $p = (p_1, \dots, p_n)$ is known, the problem (1) is a combinatorial optimization, and an offline optimization algorithm is provided in Section III. The optimal solution is denoted by S_t^*.

In reality, the probabilities of response are usually unknown. Thus, the aggregator should learn the probabilities from the feedback of previous demand response events, then make online decisions to minimize the difference between the total demand reduction and the target D. The learning performance is measured by Regret(T), which compares the total expected cost of online decisions and the optimal total expected costs in T time steps[3]:

$$\text{Regret}(T) \coloneqq \mathbb{E}[\sum_{t=1}^{T} R_t(S_t)] \tag{2}$$

where $R_t(S_t) \coloneqq L_t(S_t) - L_t(S_t^*)$ and the expectation is taken with respect to random $X_{t,i}$ and possibly random S_t.

The feedback of previous demand response events includes the responses of every selected customer, i.e., $\{X_{t,i}\}_{i \in S_t}$. Such feedback structure is called *semi-bandit* in literature

[1]The specific definition of DR event and the duration of each event is up to the choice of the system designer. Our methods can accommodate different scenarios.

[2]For simplicity, we only consider that each customer has one unit to reduce. Our method can be easily extended to multi-unit setting and/or the setting where different users have different size of units. But the regret analysis will be more complicated which we leave as future work.

[3]Strictly speaking, this is the definition of pseudo-regret, because its benchmark is the optimal expected cost: $\min_{S_t \subseteq [n]} \mathbb{E}\, L_t(S_t)$, instead of the optimal cost for each time, i.e. $\min_{S_t \subseteq [n]} L_t(S_t)$.

[13], and carries more information than bandit feedback which only includes the realized cost $L_t(S_t)$.

Lastly, we note that our problem formulation can be applied to other applications beyond demand response. One example is introduced below.

Example 1. *Consider a crowd-sourcing related problem. Given a fixed budget D, a survey planner sends out surveys and offers one unit of reward for each participant. Each potential participant may participate with probability p_i. Let $X_{t,i} = 1$ if agent i participates; and $X_{t,i} = 0$, if agent i ignores the survey. The survey planner wants to maximize the total number of responses without exceeding the budget too much. One possible formulation is to select subset S_t such that the total number of responses is close to the budget D,*

$$\min_{S_t}(\sum_{i\in S_t} X_{t,i} - D)^2$$

Since the participation probabilities are unknown, the survey planner can learn the participation probabilities from the previous actions of its selected agents and then try to minimize the total costs during the learning process.

III. Algorithm Design

In this section, we first analyze the offline optimization problem and provide an optimization algorithm. Then we introduce the notations for online algorithm analysis, and discuss two simple algorithms: greedy algorithm and CUCB (Combinatorial Upper Confidence Bound). Finally, we present our online algorithm CUCB-Avg, and provide intuitions behind the algorithm design.

A. Offline Optimization

When the probability profile p is known, the problem (1) becomes a combinatorial optimization problem:

$$\min_{S\subseteq[n]} \mathbb{E}\, L(S) \Leftrightarrow \min_{S\subseteq[n]} (\sum_{i\in S} p_i - D)^2 + \sum_{i\in S} p_i(1-p_i) \quad (3)$$

Though combinatorial optimization is NP-hard in general and only has approximate algorithms, the problem (3) admits a simple optimal algorithm, as shown in Algorithm 1. Roughly speaking, Algorithm 1 takes two steps: i) rank the arms according to p_i, ii) determine the number k according to the probability profile p and the target D and select the top k arms. The output of Algorithm 1 is denoted by $\phi(p, D)$ which is a subset of $[n]$. In the following theorem, we show that such algorithm finds an optimal solution to (3).

Theorem 1. *For any $D > 0$, the output of Algorithm 1, $\phi(p, D)$, is an optimal solution to* (3)*.*

Proof Sketch. We defer the detailed proof to [26] and only introduce the intuition here. To solve (3), we need two things: i) the total expected contribution of S, $\sum_{i\in S} p_i$, is closed to the target D, ii) the total variance of arms in S is minimized. i) is guaranteed by Line 3 of Algorithm 1: it is easy to show that $|\sum_{i\in\phi(p,D)} p_i - D| \le 1/2$. ii) is guaranteed by only selecting arms with higher response probability, as indicated by Line 2 of Algorithm 1. The intuition is given below. Consider an arm with large parameter p_1 and two arms with smaller parameters p_2, p_3. To make analysis easier, we assume $p_1 = p_2 + p_3$. Thus replacing p_1 with p_2, p_3 will not affect the first term in (3). However,

$$p_1(1-p_1) \le p_2(1-p_2) + p_3(1-p_3)$$

by $p_1^2 = (p_2+p_3)^2 \ge p_2^2 + p_3^2$. Therefore, replacing one arm with higher response probability by two arms with lower response probabilities will only increase the variance. □

Algorithm 1: Offline optimization algorithm

1: **Inputs:** n, $p_1, \ldots, p_n$, $D > 0$
2: Rank p_i in a non-increasing order:

$$p_{\sigma(1)} \ge \cdots \ge p_{\sigma(n)}$$

3: Find the smallest $k \ge 0$ such that

$$\sum_{i=1}^{k} p_{\sigma(i)} > D - 1/2$$

or $k = n$ if

$$\sum_{i=1}^{n} p_{\sigma(i)} \le D - 1/2$$

Ties are broken randomly.
4: **Ouputs**: $\phi(p, D) = \{\sigma(1), \ldots, \sigma(k)\}$

Remark 1. *There might be more than one optimal subset. Algorithm 1 only outputs one of them.*

Corollary 1. *When $D \le 1/2$, $\phi(p, D) = \emptyset$ is optimal.*

Notice that when $D \le 1/2$, the optimal subset $\phi(p, D) = \emptyset$ does not depend on p. Therefore, in the online setting, we can always find an optimal subset for $D \le 1/2$ even without any knowledge of p.

B. Notations for Online Algorithms

Let $\bar{p}_i(t)$ denote the sample average of parameter p_i by time t (including time t), then

$$\bar{p}_i(t) = \frac{1}{T_i(t)} \sum_{\tau\in I_i(t)} X_{\tau,i}$$

where $I_i(t)$ denotes the set of times steps when arm i was selected by time t and $T_i(t) = |I_i(t)|$ denotes the number of times that arm i has been selected by time t. Let $\bar{p}(t) = (\bar{p}_1(t), \ldots, \bar{p}_n(t))$. Notice that before making decisions at time t, only $\bar{p}(t-1)$ is available.

C. Two Simple Online Algorithms: Greedy Algorithm and CUCB

In this subsection, we introduce two simple algorithms: greedy algorithm and CUCB, and explain why they perform poorly in our problem.

Greedy algorithm is initialized by selecting every arm at time $t = 1$. It then uses the sample average of each parameter $\bar{p}_i(t-1)$ as an estimation of unknown probability p_i and chooses a subset based on the offline oracle described

in Algorithm 1, i.e. $S_t = \phi(\bar{p}(t-1), D)$. The greedy algorithm is expected to perform poorly because it only exploits the current information, but fails to explore the unknown information, as demonstrated below.

Example 2. *Consider two arms with parameters $p_1 > p_2$. The goal is to select the arm with the higher parameter. Now, suppose after some time steps, we have explored the suboptimal arm 2 for enough times, such that the sample average provides a good estimation $\bar{p}_2 \approx p_2$, but haven't explored the optimal arm 1 enough so that the sample average is under-estimated: $\bar{p}_1 < \bar{p}_2 < p_1$. If we apply greedy algorithm, we will keep selecting the suboptimal arm 2 based on current information: $\bar{p}_1, \bar{p}_2$, but fails to explore arm 1's information. As a result, the regret will be $O(T)$.*

A well-known algorithm in CMAB literature that balances the exploration and exploitation is CUCB [13], [17]. Instead of using sample average $\bar{p}$ directly, CUCB modifies the sample average by adding a confidence interval radius,

$$U_i(t) = \min(\bar{p}_i(t-1) + \sqrt{\frac{\alpha \log t}{2T_i(t-1)}}, 1) \tag{4}$$

where α is a positive parameter of the algorithm. Then CUCB applies the offline oracle $S_t = \phi(U(t), D)$ where $U(t) = (U_1(t), \ldots, U_n(t))$. $U_i(t)$ is restricted to $[0,1]$ in case $U(t)$ is outside the domain of the oracle ϕ. $U_i(t)$ is also known as the upper confidence bound of p_i, and is restricted to $[0,1]$ allows the algorithm to balance exploration and exploitation, because it carries the information of both the sample average, and the number of exploration times $T_i(t-1)$. CUCB performs well in classic CMAB problems, such as maximizing the total contribution of K arms for a fixed number K [13], [17].

However, CUCB performs poorly in our problem, as demonstrated by simulations in Section V. The major problem of CUCB is the over-estimation of the arm parameter p. By choosing $S_t = \phi(U(t), D)$ based on upper confidence bounds, CUCB selects less arms than needed, which not only results in a large distance between the total load reduction and the target, but also discourages exploration.

D. Our Proposed Online Algorithm: CUCB-Avg

Based on our discussion above, we propose a new algorithm, CUCB-Avg. The novelty of our algorithm is that it utilizes both sample averages and upper confidence bounds by exploiting the structure of the offline optimal algorithm.

We note that the offline algorithm 1 selects the right subset of arms in two steps: i) rank (top) arms, ii) determine the number k of the top k arms to select. In CUCB-Avg, we use the upper confidence bound $U_i(t)$ to rank the arms in a non-increasing order. This is the same as CUCB. However, the difference is that our CUCB-Avg uses the sample average $\bar{p}_i(t-1)$ to decide the number of arms to select at time t. The details of algorithm are given in Algorithm 2.

Now we explain why the ranking rule and the selection rule of CUCB-Ave are expected to work for our problem. The ranking rule is determined by $U_i(t)$ and an arm with larger $U_i(t)$ is given a priority to be selected at time t. We note that $U_i(t)$ is the summation of two terms: the sample average $\bar{p}_i(t-1)$ and the confidence interval radius that is related to how many times the arm has been explored. Therefore, if an arm i) has a small $T_i(t-1)$ indicating that arm i has not been explored enough or ii) has a large $\bar{p}_i(t-1)$ indicating that arm i might have larger parameter p_i, then arm i tends to have a larger $U_i(t)$ and thus is given a priority to be selected. In this way, CUCB-Avg selects both under-explored arms (*exploration*) and arms with large parameters (*exploitation*).

When determining k, CUCB-Avg uses the sample averages and selects enough arms such that the total sample average is close to D. Compared with CUCB which uses upper confidence bounds to determine k, our algorithm selects more arms, which reduces the distance between the total reduction and the target, and also encourages exploration.

Algorithm 2: CUCB-Avg

1: **Notations:** $T_i(t)$ is the number of times selecting arm i by time t, and $\bar{p}_i(t)$ is the sample average of arm i by time t (both including time t).
2: **Inputs:** $\alpha > 2$, D
3: **Initialization:** At $t = 1$, play $S_1 = [n]$, compute $T_i(1)$, $\bar{p}_i(1)$ according to the observation $\{X_{1,i}\}_{i \in [n]}$
4: **for** $t = 2, \ldots, T$ **do**
5: Compute the upper confidence bound for each i:

$$U_i(t) = \min(\bar{p}_i(t-1) + \sqrt{\frac{\alpha \log t}{2T_i(t-1)}}, 1)$$

6: Rank $U_i(t)$ by a non-increasing order: $U_{\sigma(t,1)}(t) \geq \cdots \geq U_{\sigma(t,n)}(t)$.
7: Find the smallest $k_t \geq 0$ such that

$$\sum_{i=1}^{k_t} \bar{p}_{\sigma(t,i)}(t-1) > D - 1/2$$

or $k_t = n$ if $\sum_{i=1}^{n} \bar{p}_{\sigma(t,i)}(t-1) \leq D - 1/2$.
8: Play $S_t = \{\sigma(t,1), \ldots, \sigma(t,k_t)\}$. Update $T_i(t)$ and $\bar{p}_i(t)$ according to the observation $\{X_{t,i}\}_{i \in S_t}$
9: **end for**

IV. Regret analysis

In this section, we will prove that our algorithm CUCB-Avg achieves $O(\log T)$ regret when D is time invariant.

The next theorem upper bounds the regret of CUCB-Avg.

Theorem 2. *Consider n arms with parameter $p = (p_1, \ldots, p_n)$ over T time steps. There exists a constant $\epsilon_0 > 0$ determined by p and D, such that for any $\alpha > 2$, the regret of CUCB-Avg is upper bounded by*

$$\text{Regret}(T) \leq M(1 + \frac{2n}{\alpha - 2}) + \frac{\alpha M n \log T}{2\epsilon_0^2} \tag{5}$$

where $M = \max(D^2, (n-D)^2)$.

Due to the space limit, we defer the proof to [26] and make a few comments below.

Regret Bound. The regret bound in (5) is $O(n^3 \log T)$ because $M \sim O(n^2)$ and ϵ_0 is a constant determined by p and D. The bound is referred as *distribution-dependent bound* in literature as p is invariant with horizon T [12].

Choice of α. α shows up in two terms: $\frac{2Mn}{\alpha-2}$ and $\frac{\alpha M n \log T}{2\epsilon_0^2}$. The first term grows when α decreases, and the second one decreases when α decreases. Since the second term is $O(\log T)$ while the first term is constant with respect to T, α should be chosen to be close to 2 when T is large.

Role of ϵ_0. We defer the explicit expression of ϵ_0 to [26] and only explain the intuition behind ϵ_0 here. To start with, we explain why the upper bound in (5) decreases when ϵ_0 increases. Roughly speaking, ϵ_0 is a robustness measure of our offline optimal algorithm, in the sense that if the probability profile p is perturbed to be $\bar{p}$ by ϵ_0 (i.e., $|\bar{p}_i - p_i| < \epsilon_0$ for all i), Algorithm 1's output $\phi(\bar{p}, D)$ would still be optimal for the true profile p. Intuitively, if ϵ_0 is large, the learning task is easy because we are able to find an optimal subset given a poor estimation, leading to a small regret.

To give a rough idea of what factors will affect the robustness measure ϵ_0, we provide an explicit expression of ϵ_0 under two assumptions in the following proposition.

Proposition 1. *If the following two assumptions hold,*
(A1): p_i are positive and distinct: $p_{\sigma(1)} > \cdots > p_{\sigma(n)} > 0$
(A2): There exists $k \geq 1$ such that

$$\sum_{i=1}^{k} p_{\sigma(i)} > D - 1/2$$
$$\sum_{i=1}^{k-1} p_{\sigma(i)} < D - 1/2$$

then the ϵ_0 in Theorem 2 can be determined by:

$$\epsilon_0 = \min\left(\frac{\delta_1}{k}, \frac{\delta_2}{k}, \frac{\Delta_k}{2}\right) \tag{6}$$

where

$$\begin{aligned} & k = |\phi(p, D)| \\ & \sum_{i=1}^{k} p_{\sigma(i)} = D - 1/2 + \delta_1, \\ & \sum_{i=1}^{k-1} p_{\sigma(i)} = D - 1/2 - \delta_2, \\ & \Delta_i = p_{\sigma(i)} - p_{\sigma(i+1)}, \forall\ i = 1, \ldots, n-1 \end{aligned} \tag{7}$$

We defer the proof to [26] and only make two comments on the proposition here. Firstly, it is easy to verify that Assumptions (A1) and (A2) imply $\epsilon_0 > 0$. Secondly, we verify that ϵ_0 defined in (6) is the robustness measure. Essentially, we need to show that if $\forall\, i,\ |\bar{p}_i - p_i| < \epsilon_0$, we have $\phi(\bar{p}, D) = \phi(p, D) \coloneqq \{\sigma(1), \ldots, \sigma(k)\}$. We prove this in two steps. Step 1: when $\epsilon_0 \leq \frac{\Delta_k}{2}$, the k arms with higher $\bar{p}_i$ are the same k arms with higher p_i because for any $1 \leq i \leq k$ and $k+1 \leq j \leq n$, we have $\bar{p}_{\sigma(i)} > p_{\sigma(k)} - \epsilon_0 \geq p_{\sigma(k+1)} + \epsilon_0 > \bar{p}_{\sigma(j)}$. Step 2: because $\epsilon_0 \leq \frac{\delta_1}{k}, \frac{\delta_2}{k}$, we have i) $\sum_{i=1}^{k} \bar{p}_{\sigma(i)} > \sum_{i=1}^{k}(p_{\sigma(i)} - \epsilon_0) = D - 1/2 + \delta_1 - k\epsilon_0 \geq D - 1/2$ and ii) $\sum_{i=1}^{k-1} \bar{p}_{\sigma(i)} < \sum_{i=1}^{k-1}(p_{\sigma(i)} + \epsilon_0) = D - 1/2 - \delta_2 + (k-1)\epsilon_0 \leq D - 1/2$ Thus, we have shown that $\phi(\bar{p}, D) = \{\sigma(1), \ldots, \sigma(k)\}$.

Finally, we briefly discuss how to generalize the expression of ϵ_0 in (6) to the case without (A1) and (A2). When (A1) does not hold, we only consider the gap between the arms that are not in a tie, i.e. $\{\Delta_i | \ \Delta_i > 0,\ 1 \leq i \leq n-1\}$. When (A2) does not hold and $\sum_{i=1}^{k-1} p_{\sigma(i)} = D - 1/2$, we consider less than $k-1$ arms to make the total expected contribution below $D - 1/2$. For the explicit expression of ϵ_0, we refer the reader to [26].

Comparsion with the regret bound of classic CMAB. In classic CMAB literature whose goal is to select K arms with highest parameters for a fixed integer K, the regret bound depends on $\frac{\Delta_K}{2}$ [17]. We note that $\frac{\Delta_K}{2}$ plays the same role as the ϵ_0 in our problem, as it is the robustness measure of the classic problem above. That is, given any estimation $\bar{p}$ with estimation error at most $\Delta_K/2$: $\forall\, i,\ |\bar{p}_i - p_i| < \Delta_K/2$, the highest K arms with the profile $\bar{p}$ are the same highest K arms with the profile p.

In addition, the regret bound in literature is $O(\frac{\log T}{\Delta_K/2})$ as Δ_K goes to zero [17], while our regret bound in (5) is $O(\frac{\log T}{\epsilon_0^2})$. This difference may be due to technical reasons.

V. Numerical Experiments

In this section, we numerically study the performance of CUCB-Avg for residential DR, and compare it with classic bandit algorithms such as CUCB and Thompson sampling [13], [21]. We will show that CUCB-Avg performs much better than CUCB and similarly to Thompson sampling.

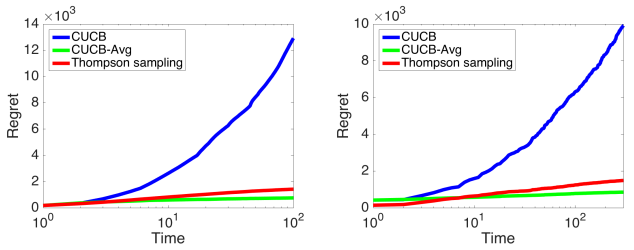
A. Thompson sampling

Thompson sampling is a Bayesian algorithm that views the unknown probability vector p as a random vector with a prior distribution. It is fundamentally different from all the algorithms mentioned above, which all view p as an unknown but deterministic vector. Thompson sampling is well-known for its good empirical performance in classical CMAB problems [22], [23], [25], thus it is worth comparing our algorithm with Thompson sampling by simulation for our problem. The theoretical analysis of Thompson sampling is both limited and complicated, thus we leave for future work the regret analysis of Thompson sampling for our problem.

For the reader's convenience, we briefly explain the algorithm procedures here. Thompson sampling first selects the subset S_t based on sample $\hat{p}_t$ from the prior distribution at $t = 1$ (or the posterior distribution at $t \geq 2$) and the offline oracle ϕ: $S_t = \phi(\hat{p}_t, D)$, then updates the posterior distribution by the feedback from the selected subset, $\{X_{t,i}\}_{i \in S_t}$. For more details, we refer the reader to [21].

B. Residential Demand Response

This section studies a residential DR program in two scenarios, 1) given a time-invariant load-reduction target, 2) given a time-varying target. Specifically, we consider $n = 100$ customers, whose response probability p_i is uniformly randomly drawn from $[0, 1]$ for all i. In the time-invariant case, let the load-reduction target be $D = 35$ units, and the time horizon be $T = 100$, and set the algorithm parameter



(a) Static case (b) Time-varying case

Fig. 1: The figures compare the regret of CUCB, CUCB-Avg and Thompson sampling. In both cases, CUCB-Avg has $O(\log T)$ regret, greatly outperforms CUCB, and performs similarly to Thompson sampling.

as $\alpha = 2.1$. In the time-varying case, let the time horizon be $T = 300$, consider target D_t to be independently uniformly drawn from $[10, 30]$, and set the algorithm parameter as $\alpha = 2.05$.

Figure 1a plots the regret of CUCB, CUCB-Avg and Thompson sampling under a time-invariant target with a logarithmic scale for the x-axis based on 200 independent simulations. The prior distribution of p is chosen to be the uniform distribution on $[0, 1]^n$. Firstly, the figure shows that the regret of CUCB-Avg is linear with respect to $\log(T)$, which matches our theoretical result in Theorem 2. In addition, the classic algorithm CUCB performs poorly in our problem, generating regret almost linear in T. This is aligned with our intuition in Section III. Finally, the figure shows that CUCB-Avg and Thompson sampling have similar performance. In this scenario, CUCB-Avg performs slightly better, but we note that there exist other scenarios where Thompson sampling is slightly better.

Figure 1b plots the regret of CUCB, CUCB-Avg and Thompson sampling under a time-varying target. Interestingly, the figure shows that CUCB-Avg still guarantees $O(\log(T))$ regret in the time-varying case, and we leave it as our future work to provide a theoretical explanation for this observation.

VI. Conclusion

In this paper, we study a combinatorial multi-armed bandit problem motivated by residential demand response with the goal of minimizing the difference between the total load adjustment and a target value. We propose a new algorithm CUCB-Avg, and show that when the target is time-invariant, CUCB-Avg achieves $O(\log T)$ regret. The numerical results also confirm the performance of the algorithm. Future work includes 1) studying the performance guarantee of Thompson sampling, 2) deriving the lower bound of the regret for our problem, 3) generalizing the model to handle dynamic target and population, and other load reduction models of customers, e.g. continuous distribution, Markov processes.

References

[1] P. Siano, "Demand response and smart grid – a survey," *Renewable and Sustainable Energy Reviews*, vol. 30, no. C, pp. 461–478, 2014.

[2] M. H. Albadi and E. F. El-Saadany, "Demand response in electricity markets: An overview," in *Power Engineering Society General Meeting, 2007. IEEE*, June 2007, pp. 1–5.

[3] N. Li, L. Chen, and S. H. Low, "Optimal demand response based on utility maximization in power networks," in *2011 IEEE power and energy society general meeting*. IEEE, 2011, pp. 1–8.

[4] "PJM: Demand response," http://www.pjm.com/markets-and-operations/demand-response.aspx, 2018.

[5] "NYISO demand response program," http://www.nyiso.com/public /markets_operations/market_data/demand_response/index.jsp.

[6] F. Rahimi and A. Ipakchi, "Demand response as a market resource under the smart grid paradigm," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 82–88, 2010.

[7] Y. Li and N. Li, "Mechanism design for reliability in demand response with uncertainty," in *American Control Conference (ACC), 2017*. IEEE, 2017, pp. 3400–3405.

[8] "Reports on Demand Response and Advanced Metering ," Federal Energy Regulatory Commission, Tech. Rep., 12 2017.

[9] D. O'Neill, M. Levorato, A. Goldsmith, and U. Mitra, "Residential demand response using reinforcement learning," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE, 2010, pp. 409–414.

[10] "Electric power monthly," https://www.eia.gov/electricity/ monthly/epm_table_grapher.php?t=epmt_5_01, 2018.

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

[12] S. Bubeck, N. Cesa-Bianchi *et al.*, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends® in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[13] W. Chen, Y. Wang, Y. Yuan, and Q. Wang, "Combinatorial multi-armed bandit and its extension to probabilistically triggered arms," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1746–1778, 2016.

[14] R. Combes, M. S. T. M. Shahi, A. Proutiere *et al.*, "Combinatorial bandits revisited," in *Advances in Neural Information Processing Systems*, 2015, pp. 2116–2124.

[15] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 5, pp. 1466–1478, 2012.

[16] B. Kveton, Z. Wen, A. Ashkan, H. Eydgahi, and B. Eriksson, "Matroid bandits: Fast combinatorial optimization with learning," *arXiv preprint arXiv:1403.5045*, 2014.

[17] B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvari, "Tight regret bounds for stochastic combinatorial semi-bandits," in *Artificial Intelligence and Statistics*, 2015, pp. 535–543.

[18] Q. Wang, M. Liu, and J. L. Mathieu, "Adaptive demand response: Online learning of restless and controlled bandits," in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*. IEEE, 2014, pp. 752–757.

[19] A. Lesage-Landry and J. A. Taylor, "The multi-armed bandit with stochastic plays," *IEEE Transactions on Automatic Control*, 2017.

[20] S. Jain, B. Narayanaswamy, and Y. Narahari, "A multiarmed bandit incentive mechanism for crowdsourcing demand response in smart grids." in *AAAI*, 2014, pp. 721–727.

[21] D. Russo, B. Van Roy, A. Kazerouni, and I. Osband, "A tutorial on thompson sampling," *arXiv preprint arXiv:1707.02038*, 2017.

[22] S. Agrawal and N. Goyal, "Analysis of thompson sampling for the multi-armed bandit problem," in *Conference on Learning Theory*, 2012, pp. 39–1.

[23] O. Chapelle and L. Li, "An empirical evaluation of thompson sampling," in *Advances in neural information processing systems*, 2011, pp. 2249–2257.

[24] J.-Y. Audibert, S. Bubeck, and G. Lugosi, "Regret in online combinatorial optimization," *Mathematics of Operations Research*, vol. 39, no. 1, pp. 31–45, 2013.

[25] A. Gopalan, S. Mannor, and Y. Mansour, "Thompson sampling for complex online problems," in *International Conference on Machine Learning*, 2014, pp. 100–108.

[26] Y. Li, Q. Hu, and N. Li. (2018) Learning and targeting the right customers for reliability: A multi-armed bandit approach (extended version). [Online]. Available: https://nali.seas.harvard.edu/files/nali/files/2018cdcmab.pdf