Learning and Selecting Users for Achieving Reliability: A Multi-armed Bandit Approach

Yingying Li [a], Qinran Hu [b], Na Li [a]

[a] *Harvard University, Cambridge, MA 02138, USA.*

[b] *Southeast University, Nanjing, China*

Abstract

One challenge in the optimization and control of societal systems is to handle the unknown and uncertain user behavior. This paper focuses on residential demand response (DR) and proposes a closed-loop learning scheme to address these issues. In particular, we consider DR programs where an aggregator calls upon residential users to change their demand so that the total load adjustment is close to a target value. To learn and select the right customers, we formulate the DR problem as a combinatorial multi-armed bandit (CMAB) problem with a reliability objective. 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 consider both a fixed time-invariant target and time-varying targets and show that CUCB-Avg achieves $O(\log T)$ and $O(\sqrt{T\log(T)}$ regrets respectively. We further numerically test our algorithms using synthetic realistic DR data and demonstrate that our CUCB-Avg performs significantly better than the classic CUCB and better than Thompson Sampling.

Key words: learning theory; optimization under uncertainties; real time simulation and dispatching; multi-armed bandit; demand response; regret analysis.

1 Introduction

In many sequential decision-making problems of societal systems such as transportation, electricity grids, communication, crowd-sourcing, and resource allocation problems in general, one key challenge is to handle the unknown and uncertain user behavior. This paper focuses on residential demand response and aims to design a closed-loop learning scheme to address this challenge. Demand response (DR) has been playing an increasing role in electricity markets for improving grid resilience and sustainability [1–7]. However, most of the existing successful DR programs are for commercial and industrial customers. Despite a growing effort, residential DR remains underexploited [6]. In most residential DR programs, an aggregator such as a utility company requests load changes from users, for example, by changing the temperature set points of air conditioners. To encourage users' participation, most programs use incentive schemes such as prices, rewards, coupons, raffles, etc, under the assumption that customers are price responsive [2,6–8]. However, because the average monetary reward budget for a single household is usually small, it is reported that rewards play a limited role in affecting users' DR behaviors [6]. On the other side, there are many other factors that affect residential DR decisions, such as house sizes and types, household demographics and lifestyles, humidity and temperature, etc. However, it is unclear how these factors quantitatively affect DR actions. People with similar factors might react to a same DR signal in very different ways. Moreover, the DR aggregator usually has limited knowledge of these factors. These intrinsic, heterogeneous uncertainties associated with residential customers call for learning approaches to understand and interact with residential customers in a delicate way.

Multi-armed bandit (MAB) emerges as a natural framework to handle such uncertainties [9,10]. In a simple setting, MAB considers n independent arms, each providing a random contribution according to its own distribution at time steps $1 \leq t \leq T$. Without knowing these distributions, a decision maker picks one arm at each time

⋆ The work was supported by NSF CAREER 1553407,NSF ECCS 1839632, AFOSR YIP, ONR YIP, and ARPA-E through the NODES program.

Email addresses: yingyingli@g.harvard.edu (Yingying Li), qhu@seu.edu.cn (Qinran Hu), nali@seas.harvard.edu (Na Li).

step, and tries to maximize the total expected contribution. The decision maker decides whether to *explore* arms to learn the unknown distributions, or to *exploit* the current knowledge by selecting the arm that has provided the highest mean contribution. When the decision maker can select multiple arms at each time, the problem is referred to as CMAB (Combinatorial MAB) in literature [11–15]. (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 regret, which captures the difference between the optimal expected value assuming the distributions are known and the achieved expected 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 have been studies of DR via (C)MAB [16–18,7]. However, most literature sets the goal as maximizing the load reduction for peak hours rather than following a load reduction target for reliability issues. Besides, conducting rigorous regret analysis for achieving sub-linear regret is always challenging.

1.1 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 signal. The target might due to a sudden change of renewable energy or a peak load reduction request. for the sake of grid 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 DR aggregator is to select a subset of the customers to approximate the target as close as possible. The size of the subset is not fixed, giving flexibility to the aggregator for achieving different levels of targets. Compared with the classic CMAB literature [11–15], a major difference of our formulation is that the reliability objective leads to a non-monotonic objective making the existing CMAB approaches and regret analysis inapplicable here [11].

In order to design our CMAB online learning algorithm, we first study the corresponding offline combinatorial optimization problem assuming the probabilities of the load change are known. Based on the structure of this 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 and $O(\sqrt{T\log(T)})$ regret given a time-varying target. The dependence of regret on the dimension n (the number of customers) is polynomial in both cases. We also conduct numerical studies using synthetic DR data, showing that the performance of CUCB-Avg is much better than the classic algorithm CUCB [11,14,15] and slightly better than Thompson sampling, another popular CMAB method which uses a Bayesian learning approach and has good empirical performance [19–21].

Lastly, we would like to point out that despite a simplified DR model used in this paper, the model was motivated by real-world field studies of residential DR and the results in this paper have served as a guideline to design learning protocols for these studies [22]. Moreover, through modifying the detailed algorithms for other applications, the overall closed-loop learning scheme and the technical tools for analyzing its theoretical performance are useful for other societal systems especially with human-in-the-loop.

1.2 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) [10,15,14,13,12,23].

As for more general settings, Chen et. al. design Combinatorial Upper Confidence Bound (CUCB) using the principle of *optimism in the face of uncertainty* when the objective function is *monotonically nondecreasing* with the parameters of arms given a fixed selected subset [11]. However, the reliability objective of our CMAB problem does not satisfy the monotone conditions. Another line of work follows the *Bayesian* approach and studies Thompson sampling [24,25]. However, the regret bounds of Thompson sampling depend exponentially on the number of the selected arms K and [25] shows that the exponential dependence is unavoidable. In the residential DR problems, K is usually large, thus Thompson sampling may generate poor performance. Finally, there is a lack of analysis on time-varying objective functions, but in many real-world applications the objectives change with time. For example, in residential DR, the load reduction targets would depend on the time-varying renewable generation. Therefore, either the learning algorithms or the theoretical analysis in literature do not directly apply to our CMAB problem, motivating the work of this paper.

Some of the preliminary work was presented in the conference paper [26]. This journal version strengthens the regret bounds, especially for the time-varying target case, conducts much more intensive numerical analysis using realistic data from ISOs, provides more complete proofs, and adds more physical intuitions and discussions on both theoretical and numerical results.

Notations. Let $\bar{E}$ and $|E|$ be the complement and the cardinality of the set E respectively. For any positive integer n, let $[n] = \{1, \ldots, n\}$. Let $I_E(x)$ be the indicator function: $I_E(x) = 1$ if $x \in E$ and $I_E(x) = 0$ if $x \notin E$.

For any two sets A, B, $A - B \coloneqq \{x \mid x \in A, x \notin B\}$. When $k = 0$, let $\sum_{i=1}^{k} a_i = 0$ for any a_i, and the set $\{\sigma(1), \ldots, \sigma(k)\} = \emptyset$ for any $\sigma(i)$. For $x \in \mathbb{R}^k$, 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 $f(x) = o(g(x))$ if $\lim_{x\to+\infty} f(x)/g(x) = 0$. We usually omit "as $x \to +\infty$" for simplicity. For the asymptotic behavior near zero, consider the inverse of x.

2 Problem Formulation

Motivated by the discussion in the introduction, we formulate the DR as a CMAB problem in this section. We focus on 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 with probability $1 - p_i$. 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 $D_t \geq 0$ 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_t:

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

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

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

In this paper, we will first study the scenario where the target D is time-invariant (Section 3 and 4). Then we will extend the results to cope with time-varying targets to incorporate different DR signals resulted from the fluctuations of power supply and demand (Section 5).

[1] The specific definition of DR events 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 learning method can be 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. As mentioned before, results in the paper have been used as a guideline for DR field studies[22].

When the response probability profile $p = (p_1, \ldots, p_n)$ is known, the problem (1) is a combinatorial optimization, and an offline optimization algorithm is provided in Section 3. Let S_t^* denote an optimal solution.

In reality, the response probabilities 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_t. The learning performance is measured by $\text{Regret}(T)$, 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 $X_{t,i}$ and the 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 [11], 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 budget D_t, 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_t,*

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

Since the participation probabilities are unknown, the 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.

3 Algorithm Design

This section considers time-invariant target D. We will first provide an optimization algorithm for the offline

[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. $\min_{S_t \subseteq [n]} L_t(S_t)$.

problem, then introduce the notations for online algorithms and discuss two simple algorithms: greedy algorithm and CUCB. Finally, we introduce our online algorithm CUCB-Avg.

3.1 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 and only has approximate algorithms in general, we are able to design a simple algorithm in Algorithm 1 to solve the problem (3) exactly. 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).

Algorithm 1 Offline optimization algorithm

1: **Inputs:** $p_1, \ldots, p_n \in [0,1]$, $D > 0$.
2: Rank p_i in a non-increasing order: $p_{\sigma(1)} \geq \cdots \geq p_{\sigma(n)}$.
3: Find the smallest $k \geq 0$ such that

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

Let $k = n$ if $\sum_{i=1}^{n} p_{\sigma(i)} \leq D - 1/2$. Ties are broken randomly.
4: **Ouputs**: $\phi(p, D) = \{\sigma(1), \ldots, \sigma(k)\}$

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 [27] and only introduce the intuition here. An optimal set S roughly has two properties: 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 roughly guaranteed by Line 3 of Algorithm 1: it is easy to show that $|\sum_{i\in\phi(p,D)} p_i - D| \leq 1/2$. ii) is roughly guaranteed by only selecting arms with higher response probabilities, as indicated by Line 2 of Algorithm 1. The intuition is the following. Consider an arm with large parameter p_1 and two arms with smaller parameters p_2, p_3. For simplicity, we let $p_1 = p_2 + p_3$. Thus replacing p_1 with p_2, p_3 will not affect the first term in (3). However, $p_1(1-p_1) \leq p_2(1-p_2) + p_3(1-p_3)$ by $p_1^2 = (p_2+p_3)^2 \geq p_2^2 + p_3^2$. Hence, replacing one arm with higher response probability by two arms with lower response probabilities will increase the variance. □

Corollary 1 *When $D < 1/2$, the empty set is optimal.*

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

3.2 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 time steps when arm i is selected before time t and $T_i(t) = |I_i(t)|$ denotes the number of times that arm i has been selected before 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.

3.3 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 to gain intuitions for our algorithm development.

Greedy algorithm uses the sample average of each parameter $\bar{p}_i(t-1)$ as an estimation of the 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 known 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. Suppose after some time steps, we have explored the suboptimal arm 2 for enough times such that $\bar{p}_2 \approx p_2$, but haven't explored the optimal arm 1 enough so that $\bar{p}_1$ is an underestimate and satisfies: $\bar{p}_1 < \bar{p}_2 < p_1$. In this case, the greedy algorithm will keep selecting the suboptimal arm 2 based on current estimation $\bar{p}_1, \bar{p}_2$ and fail to explore arm 1. As a result, the regret will be $O(T)$.*

A well-known algorithm in CMAB literature that balances the exploration and exploitation is CUCB [11,15]. Instead of using sample average $\bar{p}$ directly, CUCB considers an upper confidence bound:

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

where $\alpha \geq 0$ is the parameter to balance the tradeoff between $\bar{p}_i(t-1)$ (exploitation) and $T_i(t-1)$ (exploration). The output of CUCB is $S_t = \phi(U(t), D)$. CUCB performs well in classic CMAB problems, such as maximizing the total contribution of K arms for a fixed K.

However, CUCB performs poorly in our problem, as shown in Section 6. The major problem of CUCB is the over-estimate of the arm parameter p. By choosing $S_t = \phi(U(t), D)$, CUCB selects less arms than needed,

which not only results in a large deviation from the target, but also discourages exploration.

Algorithm 2 CUCB-Avg

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).
Inputs: $\alpha > 2$, D
Initialization: For $t = 1, \ldots, \lceil \frac{n}{\lceil 2D \rceil} \rceil$, select $\lceil 2D \rceil$ arms each time until each arm has been selected for at least once. Let S_t be the set of arms selected at time t. Initialize $T_i(t)$, $\bar{p}_i(t)$ by the observation $\{X_{t,i}\}_{i \in S_t}$. [4]
for $t = \lceil \frac{n}{\lceil 2D \rceil} \rceil + 1, \ldots, T$ **do**
  Compute the upper confidence bound for each i $U_i(t) = \min(\bar{p}_i(t-1) + \sqrt{\frac{\alpha \log t}{2T_i(t-1)}}, 1)$.
  Rank $U_i(t)$ by a non-increasing order: $U_{\sigma(t,1)}(t) \geq \cdots \geq U_{\sigma(t,n)}(t)$.
  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 $\sum_{i=1}^{n} \bar{p}_{\sigma(t,i)}(t-1) \leq D - 1/2$.
  Select $S_t = \{\sigma(t,1), \ldots, \sigma(t,k_t)\}$ and update $T_i(t)$ and $\bar{p}_i(t)$ based on the observations $\{X_{t,i}\}_{i \in S_t}$
end for

3.4 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 the algorithm are given in Algorithm 2.

Now we explain why the ranking rule and the selection rule of CUCB-Ave would work for our problem.

The ranking rule is determined by $U_i(t)$. 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, an arm with a large $U_i(t)$ may either has a small $T_i(t-1)$, meaning that the arm has not been explored enough or has a large $\bar{p}_i(t-1)$, indicating that the arm frequently responds in the history. In this way, CUCB-Avg selects both the under-explored arms (*exploration*) and the arms with good performance in the past (*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 load reduction difference from the target and also encourages exploration.

[4] The initialization method is not unique and can be any method that selects each customer for at least once.

4 Regret analysis

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

4.1 The Main Result

Theorem 2 *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*

$$\text{Regret}(T) \leq M(\lceil \frac{n}{\lceil 2D \rceil} \rceil + \frac{2n}{\alpha - 2}) + \frac{\alpha M n \log T}{2\epsilon_0^2} \quad (5)$$

where $M = \max(D^2, (n-D)^2)$. □

We make a few comments before the proof.

Dependence on T and n. The dependence on the horizon T is $O(\log T)$, so the average regret diminishes to zero as T increases, demonstrating that our algorithm learns the customers' response probabilities effectively. The dependence on the number of arms n is polynomial $O(n^3)$ since $M \sim O(n^2)$, demonstrating that our algorithm can effectively learn a large number of customers.

Role of ϵ_0. The bound depends on a constant term ϵ_0 determined by p and D and such a bound is referred to as a *distribution-dependent bound* in literature. We defer the explicit expression of ϵ_0 to [27] and only explain the intuition behind ϵ_0 here. 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), the output $\phi(\bar{p}, D)$ of Algorithm 1 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. This explains why the upper bound in (5) decreases when ϵ_0 increases.

To discuss 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: $p_{\sigma(1)} > \cdots > p_{\sigma(n)} > 0$; and (A2): There exists $k \geq 1$ such that $\sum_{i=1}^{k} p_{\sigma(i)} >$*

$D - 1/2$, *and* $\sum_{i=1}^{k-1} p_{\sigma(i)} < D - 1/2$, *then the* ϵ_0 *in Theorem 2 can be determined by:*

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

where $k = |\phi(p, D)|$, $\sum_{i=1}^{k} p_{\sigma(i)} = D - 1/2 + \delta_1$, $\sum_{i=1}^{k-1} p_{\sigma(i)} = D - 1/2 - \delta_2$, *and* $\Delta_i = p_{\sigma(i)} - p_{\sigma(i+1)}, \forall\, i = 1, \ldots, n-1$.

We defer the proof of the proposition to our online report [27] and only make two comments here. Firstly, it is easy to verify that Assumptions (A1) and (A2) imply $\epsilon_0 > 0$. Secondly, we explain why ϵ_0 defined in (6) is a robustness measure. That is, if $\forall\, i,\ |\bar{p}_i - p_i| < \epsilon_0$, we have $\phi(\bar{p}, D) = \phi(p, D)$. This can be proved 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) $\sum_{i=1}^{k-1} \bar{p}_{\sigma(i)} < \sum_{i=1}^{k-1} (p_{\sigma(i)} + \epsilon_0) = D - 1/2 - \delta_2 + (k-1)\epsilon_0 \leq D - 1/2$ Therefore, by Algorithm 1, $\phi(\bar{p}, D) = \{\sigma(1), \ldots, \sigma(k)\} = \phi(p, D)$.

Finally, we briefly discuss how to generalize the expression (6) of ϵ_0 to cases 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 $D - 1/2$. Again, an explicit expression of ϵ_0 is provided in our report [27].

Comparison with the regret bound of classic CMAB. In classic CMAB literature when the goal is to select K arms with highest parameters given a fixed integer K, the regret bound usually depends on $\frac{\Delta_K}{2}$ [15]. We note that $\frac{\Delta_K}{2}$ is similar to ϵ_0 in our problem, as it is the robustness measure of the top-K-arm problem in the sense that given any estimation $\bar{p}$ with estimation error at most $\frac{\Delta_K}{2}$: $\forall\, i,\ |\bar{p}_i - p_i| < \frac{\Delta_K}{2}$, the highest K arms with the profile $\bar{p}$ are the same as that with the profile p. In addition, we would like to mention that the regret bound in literature is usually linear on $1/\Delta_K$, while our regret bound is $1/\epsilon_0^2$. This difference may be an artificial effect of the proof techniques because our CMAB problem is more complicated. We leave it as future work to strengthen the results.

Choice of α. When α increases, the term $\frac{2Mn}{\alpha - 2}$ decreases while the term $\frac{\alpha M n \log T}{2\epsilon_0^2}$ increases. Since the second term is $O(\log T)$ while the first term is $O(1)$, α should be close to 2 when T is large.

4.2 Proof of Theorem 2

Proof outline: We divide the T time steps into four parts, and bound the regret in each part separately. The partition of the time steps are based on event E_t and the event $B_t(\epsilon_0)$ defined below. Let E_t be the event when the sample average is outside the confidence interval considered in Algorithm 2:

$$E_t := \{\exists\, i \in [n],\ |\bar{p}_i(t-1) - p_i| \geq \sqrt{\frac{\alpha \log t}{2T_i(t-1)}}\}$$

For any $\epsilon > 0$, let $B_t(\epsilon)$ denote the event when Algorithm 2 selects an arm who has been explored for no more than $\frac{\alpha \log T}{2\epsilon^2}$ times:

$$B_t(\epsilon) := \{\exists\, i \in S_t,\ s.t.\ T_i(t-1) \leq \frac{\alpha \log T}{2\epsilon^2}\} \tag{7}$$

Let $\epsilon_0 > 0$ be a small number such that Lemma 3 holds.

Now, we will define the four parts of the T time steps, and briefly introduce the regret bound of each part.

(1) *Initialization:* the regret bound does not depend on T (Inequality (8)).
(2) *When E_t happens:* the regret bound does not depend on T because E_t happens rarely due to concentration properties in statistics (Lemma 1).
(3) *When $\bar{E}_t$, $B_t(\epsilon_0)$ happen:* the regret is at most $O(\log T)$ because $B_t(\epsilon_0)$ happens for at most $O(\log T)$ times (Lemma 2).
(4) *When $\bar{E}_t$, $\bar{B}_t(\epsilon_0)$ happen,* no regret due to enough exploration of the selected arms (Lemma 3).

Notice that the time steps are not divided sequentially here. For example, it is possible that $t = 10, 30$ belong to Part 2 while $t = 10$ belongs to Part 3.

Proof details: Firstly, it is without loss of generality to require $D \geq 1/2$ because when $D < 1/2$, the optimal set is empty and there is no regret by Corollary 1.

Secondly, we note that for all time steps $1 \leq t \leq T$ and any $S_t \subseteq [n]$, the regret at t is upper bounded by

$$R_t(S_t) \leq L_t(S_t) \leq \max(D^2, (n-D)^2) =: M \tag{8}$$

Thus, the regret of initialization (Part 1) at $t = 1, \ldots, \lceil \frac{n}{\lceil 2D \rceil} \rceil$ is bounded by $M \lceil \frac{n}{\lceil 2D \rceil} \rceil$.

Next, we bound the regret of Part 2 by the Chernoff-Hoeffding's concentration inequality. The intuition behind the proof is that E_t happens rarely because the sample average $\bar{p}_i(t)$ concentrates around the true value p_i with a high probability.

Theorem 3 (Chernoff-Hoeffding's inequality) *$X_1, \ldots, X_m$ i.i.d in $[0,1]$ with mean μ, then*

$$\mathbb{P}(|\sum_{i=1}^{m} X_i - m\mu| \geq m\epsilon) \leq 2e^{-2m\epsilon^2} \tag{9}$$

Lemma 1 *When $\alpha > 2$, $\mathbb{E} \sum_{t=1}^{T} I_{E_t} R_t(S_t) \leq \frac{2Mn}{\alpha - 2}$.*

Proof. The number of times E_t happens is bounded by

$$\begin{aligned}
&\mathbb{E}\sum_{t=1}^{T} I_{E_t} = \sum_{t=1}^{T}\mathbb{P}(E_t)\\
&\le \sum_{t=1}^{T}\sum_{i=1}^{n}\mathbb{P}\left(|\bar{p}_i(t-1)-p_i| \ge \sqrt{\frac{\alpha\log t}{2T_i(t-1)}}\right)\\
&\le \sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{s=1}^{t-1}\mathbb{P}(|\bar{p}_i(t-1)-p_i| \ge \sqrt{\frac{\alpha\log t}{2s}}, T_i(t-1)=s)\\
&\le \sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{s=1}^{t-1}\frac{2}{t^{\alpha}} \le \sum_{t=1}^{T} n\frac{2}{t^{\alpha-1}} \le \frac{2n}{\alpha-2}
\end{aligned}$$

where the first inequality is by enumerating possible $i \in [n]$, the second inequality is by enumerating possible values of $T_i(t-1)$: $\{1,\dots,t-1\}$, the third inequality is by Chernoff-Hoeffding's inequality, and the last inequality is by $\sum_{t=1}^{T}\frac{1}{t^{\alpha-1}} \le \int_1^{+\infty}\frac{1}{t^{\alpha-1}} \le \frac{1}{\alpha-2}$. Then by inequality (8) the proof is completed. □

Next, we show the regret of Part 3 is at most $O(\log T)$.

Lemma 2 *For any $\epsilon_0 > 0$, the regret in Part 3 is bounded by $\mathbb{E}\sum_{t=1}^{T} R_t(S_t) I_{\{\bar{E}_t, B_t(\epsilon_0)\}} \le \frac{\alpha M n \log T}{2\epsilon_0^2}$*

Proof. By the definition of $B_t(\epsilon_0)$ in (7), whenever $B_t(\epsilon_0)$ happens, the algorithm selects an arm i that has not been selected for $\frac{\alpha\log T}{2\epsilon_0^2}$ times, increasing the selection time counter $T_i(t)$ by one. Hence, $B_t(\epsilon_0)$ can happen for at most $\frac{\alpha n\log T}{2\epsilon_0^2}$ times. Then, by inequality (8), the proof is completed. □

When $\bar{E}_t$ and $\bar{B}_t(\epsilon_0)$ happen (Part 4), every selected arm is fully explored and every arm's sample average is within the confidence interval. As a result, CUCB-Avg selects the right subset and hence contributes zero regret. This is formally stated in the following lemma.

Lemma 3 *There exists $\epsilon_0 > 0$, such that for each $1 \le t \le T$, if $\bar{E}_t$ and $\bar{B}_t(\epsilon_0)$ happen, CUCB-Avg selects an optimal subset and $\mathbb{E}\, R_t(S_t) I_{\{\bar{E}_t, \bar{B}_t(\epsilon_0)\}} = 0$. Consequently, the regret in Part 4 is 0.*

Proof Sketch: We defer the proof to [27] and sketch the proof ideas here, which is based on two facts:
Fact 1: when $\bar{E}_t$ and $\bar{B}(\epsilon_0)$ happen, the upper confidence bounds can be bounded by $U_i(t) > p_i$ for all $i \in [n]$, and the confidence bounds of the selected arm j satisfy

$$|\bar{p}_j(t-1) - p_j| < \epsilon_0,\ U_j(t) < p_j + 2\epsilon_0,\ \forall\, j \in S_t.$$

Fact 2: when ϵ_0 is small enough, CUCB-Avg selects an optimal subset.

To get the intuition for Fact 2, we consider the expression of ϵ_0 in (6) under Assumption (A1) (A2) in Proposition 1. Let $\phi(p, D) = \{\sigma(1),\dots,\sigma(k)\}$ denote the optimal subset. In the following, we roughly explain why the selected subset S_t is optimal given ϵ_0 defined in (6):

i) By $\epsilon_0 \le \frac{\Delta_k}{2}$, we can show that the selected subset S_t is either a superset or a subset of the optimal subset $\{\sigma(1),\dots,\sigma(k)\}$.

ii) By $\epsilon_0 \le \delta_1/k$, we can show that we will not select more than k arms, because, informally, even if we underestimate p_i, the sum of arms in $\{\sigma(1),\dots,\sigma(k)\}$ is still larger than $D - 1/2$.

iii) By $\epsilon_0 \le \delta_2/k$, we can show that we will not select less than k arms, because, informally, even if we overestimate p_i, the sum of $k-1$ arms in $\{\sigma(1),\dots,\sigma(k)\}$ is still smaller than $D - 1/2$. □

The proof of Theorem 2 is completed by summing up the regret bounds of Part 1-4.

5 Time-varying target

In practice, the load reduction target is usually time-varying. We will study the performance of CUCB-Avg in the time-varying case below.

Notice that CUCB-Avg can be directly applied to the time-varying case by using D_t in Line 5 and Line 9 in Algorithm 2 at each time step t.

Next, we provide a regret bound for CUCB-Avg in the time-varying case. Notice that we impose no assumption on D_t except that it is bounded, which is almost always the case in practice.

Assumption 1 *There exists a finite $\bar{D} > 0$ such that $0 < D_t \le \bar{D},\ \forall\, 1 \le t \le T$.*

Theorem 4 *Suppose Assumption 1 holds. For any $\alpha > 2$, the regret of CUCB-Avg is upper bounded by*

$$\begin{aligned}
\text{Regret}(T) \le\ & \bar{M}n + \frac{2\bar{M}n}{\alpha-2} + \frac{\alpha\bar{M}n\log T}{2\epsilon_1^2}\\
&+ n^2\sqrt{2\alpha\log T}\sqrt{T + \frac{\alpha\log T}{2\epsilon_1^2}}
\end{aligned}$$

where $\bar{M} = \max(\bar{D}^2, n^2)$, $\epsilon_1 = \min(\frac{\Delta_{min}}{2}, \frac{\beta_{min}}{n})$, $\Delta_{min} = \min\{\Delta_i \mid 1 \le i \le n-1, \Delta_i > 0\}$ and $\beta_{min} = \min\{p_i \mid 1 \le i \le n, p_i > 0\}$.

Before the proof, we make a few comments below.

Dependence on T. The bound is $O(\sqrt{T\log T})$, still sublinear in T, meaning that our algorithm learns the customers' response probabilities well enough to yield diminishing average regret in the time-varying case.

The dependence on T is worse than the static case which is $O(\log T)$. We briefly discuss the intuition behind this difference. In the proof of Theorem 2, we show that there exists a threshold ϵ_0 depending on D such that when the estimation errors of parameter p_i for $i \in S_t$ are below ϵ_0, our algorithm selects the optimal subset (Lemma 3).

Moreover, we also show that as t increases, with high probability the estimation error will decrease and eventually our algorithm will find the optimal subset and generate no more regret. However, in the time-varying case the argument above no longer holds because the threshold ϵ_0 will change with D_t, denoted as $\epsilon_0(D_t)$, and it is possible that the estimation error will always be larger than $\epsilon_0(D_t)$, as a result the algorithm will not find the optimal subset with high probability. This roughly explains why the bound of the time-varying case is worse than that of the static case.

Nevertheless, we are able to show that under some conditions the regret at time t is almost bounded by the estimation error at time t (Lemma 5), and the estimation error almost scales like $O(\sqrt{\log T/t})$. Therefore, even in the worst case, the total regret can be roughly bounded by $\sum_{t=1}^{T} O(\sqrt{\log T/t}) = O(\sqrt{T \log T})$.

Finally, we note that the regret bound is for the worst-case scenario and the regret in practice may be smaller.

Dependence on n. The bound is polynomial on the number of arms n: $O(n^3)$, because $\bar{M} \sim O(n^2)$, demonstrating that our algorithm can learn a large number of arms effectively in the time-varying case.

Role of ϵ_1. Notice that ϵ_1 only depends on p and does not depend on the target D_t. Roughly speaking, ϵ_1 captures how difficult it is to rank the arms correctly by the value of p_i, in the sense that as long as the estimation error of each p_i is smaller than ϵ_1, the rank based on the estimation will be the correct rank based on the true parameter p_i.

5.1 Proof of Theorem 4

Most of the proof is similar to the static case. We also divide the time steps into four parts and complete the proof by combining the regret bound of each part. The first three parts can be bounded in similar ways as the static case. The major difference comes from Part 4.

(1) Initialization: the regret can be bounded by $\bar{M}n$ because the initialization at most lasts for n time steps and $\bar{M}$ is an upper bound of the single-step regret.

(2) When E_t happens: notice that Lemma 1 still holds in the time-varying case if replacing M with $\bar{M}$, so the second part is bounded by $\mathbb{E}\sum_{t=1}^{T} I_{E_t} R_t(S_t) \leq \frac{2\bar{M}n}{\alpha-2}$.

(3) When $\bar{E}_t$, $B_t(\epsilon_1)$ happen: notice that Lemma 2 still holds so $\mathbb{E}\sum_{t=1}^{T} R_t(S_t) I_{\{\bar{E}_t, B_t(\epsilon_1)\}} \leq \frac{\alpha \bar{M} n \log T}{2\epsilon_1^2}$.

(4) When $\bar{E}_t$, $\bar{B}_t(\epsilon_1)$ happen, we can show that the regret is $O(\sqrt{T \log T})$ as stated in the lemma below.

Lemma 4

$$\mathbb{E}\sum_{t=1}^{T} R_t(S_t) I_{\{\bar{E}_t, \bar{B}_t(\epsilon_1)\}} \leq 2n^2 \sqrt{\frac{\alpha \log T}{2}} \sqrt{T + \frac{\alpha \log T}{2\epsilon_1^2}}$$

Proof. Without loss of generality, we consider $D_t \geq 1/2$ for all t, because if $D_t < 1/2$ for some t, the algorithm will select the empty set and produce no regret at t.

Below is the intuition behind the proof. First, Lemma 5 shows that the regret at time t can roughly be bounded by the estimation error ϵ at t when $\epsilon \leq \epsilon_1$. Then, we show that the estimation error, roughly captured by $\sqrt{\log T/T_i(t-1)}$, almost decays as $O(\sqrt{\log T/t})$. Thus, the total regret is bounded by $\sum_t \sqrt{\log T/t} = \sqrt{T \log T}$.

Lemma 5 *Consider any t and D_t. For any $0 < \epsilon \leq \epsilon_1$ such that $\mathbb{P}(\bar{E}_t, \bar{B}_t(\epsilon)) > 0$, let $\mathbb{S}_t$ denote the set of all possible selections of CUCB-Avg at time t if event $\bar{E}_t$ and $\bar{B}_t(\epsilon)$ happen, then $\forall\, s \in \mathbb{S}_t$, we have $\mathbb{E}[R_t(s)] \leq 2n\epsilon$.*

Proof sketch. Due to the space limit, we defer the proof to [27] and only sketch the proof here. Firstly, we are able to show that under $\bar{E}_t, \bar{B}_t(\epsilon)$, the selected subset differs from the optimal subset for at most one arm. This is mainly due to $\epsilon \leq \epsilon_1$. Secondly, we can bound the regret of the suboptimal selections by $O(\epsilon)$, which is mainly due to our quadratic loss function. □

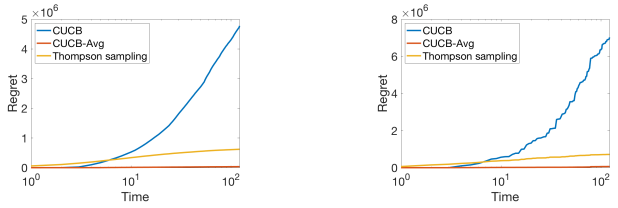
Next, we will finish the proof by bounding the estimation errors. We introduce event H_t^q to represent that each selected arm i at time t has been selected for more than $\frac{\alpha \log T}{2\epsilon_1^2} + q$ times for $q = 0, 1, 2, \ldots$:

$$H_t^q \coloneqq \{\forall\, i \in S_t, T_i(t-1) > \frac{\alpha \log T}{2\epsilon_1^2} + q\} \cap \bar{E}_t \cap \bar{B}_t(\epsilon_1)$$

In addition, we define the estimation error η_q by the confidence interval radius when an arm has been explored for $\frac{\alpha \log T}{2\epsilon_1^2} + q - 1$ times: $\frac{\alpha \log T}{2\eta_q^2} = \frac{\alpha \log T}{2\epsilon_1^2} + q - 1$, that is, $\eta_q = \sqrt{\frac{\frac{\alpha \log T}{2}}{q - 1 + \frac{\alpha \log T}{2\epsilon_1^2}}}$. The proof is completed by:

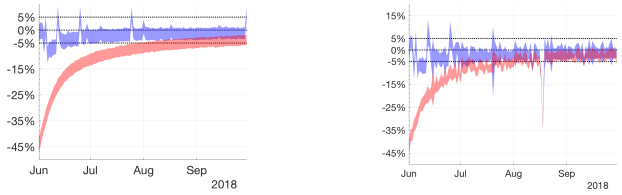
$$\begin{aligned}
&\mathbb{E}\sum_{t=1}^{T} R_t(S_t) I_{\bar{E}_t \cap \bar{B}_t(\epsilon_1)} = \sum_{q=1}^{T}\sum_{t=1}^{T} \mathbb{E}\, R_t(S_t) I_{(H_t^{q-1} - H_t^q)} \\
&\leq \sum_{q=1}^{T}\sum_{t=1}^{T} 2n\eta_q\, \mathbb{E}\, I_{(H_t^{q-1} - H_t^q)} \\
&\leq \sum_{q=1}^{T} 2n^2 \eta_q = 2n^2 \sum_{q=1}^{T} \sqrt{\frac{\frac{\alpha \log T}{2}}{q - 1 + \frac{\alpha \log T}{2\epsilon_1^2}}} \\
&\leq 2n^2 \sqrt{\frac{\alpha \log T}{2}} \left(\sqrt{T + \frac{\alpha \log T}{2\epsilon_1^2}} - \sqrt{\frac{\alpha \log T}{2\epsilon_1^2}}\right)
\end{aligned}$$

where the first equality is by $\bar{E}_t \cap \bar{B}_t(\epsilon_1) = \cup_{q=1}^{T}(H_t^{q-1} - H_t^q)$; the first inequality is by Lemma 5, $(H_t^{q-1} - H_t^q) \subseteq \bar{E}_t \cap \bar{B}_t(\eta_q)$ and $\eta_q \leq \epsilon_1$; the second one is because



(a) Average peak. (b) Daily peak.

Fig. 1. The regret of three algorithms.



(a) Average peak. (b) Daily peak.

Fig. 2. 90% confidence intervals of load reduction's relative errors of CUCB-Avg (blue) and Thompson sampling (red).

$H_t^{q-1} - H_t^q \subseteq \bigcup_{i=1}^n \{i \in S_t,\, T_i(t-1) = \frac{\alpha \log T}{2\epsilon_1^2} + q\}$ and $\{i \in S_t,\, T_i(t-1) = \frac{\alpha \log T}{2\epsilon_1^2} + q\}$ occurs at most once. □

6 Numerical Experiments

In this section, we conduct numerical experiments to complement the theoretical analysis above.

6.1 Algorithms comparison

We will compare our algorithm with Thompson sampling (TS), an algorithm with good empirical performance in classic MAB problems. We briefly introduce the Thompson sampling below. In TS, the unknown parameter p is viewed as a random vector with a prior distribution. The algorithm selects a subset $S_t = \phi(\hat{p}_t, D)$ based on a sample $\hat{p}_t$ from the prior distribution of p at $t = 1$ (or the posterior distribution at $t \geq 2$), then updates the posterior distribution of p by observations $\{X_{t,i}\}_{i \in S_t}$. For more details, we refer the reader to [19].

In our experiment, we consider a residential demand response program with 3000 customers. Each customer can either participate in the DR event by reducing 1kW or not. The probabilities of participation are i.i.d. Unif$[0, 1]$. The demand response events last for one hour on each day from June to September in 2018, with a goal of shaving the peak loads in Rhode Island. The *hourly* demand profile is from New England ISO [28]. We consider two schemes to determine the peak-load-shaving target D_t:

i) Average peak: Consider the average daily load profile in these four months. The constant target D is the 5% of the difference between the peak average load and the average load at one hour before the peak hour.
ii) Daily peak: On each day t, the target D_t is 5% of the difference between the peak load and the load at one hour before the peak hour of the daily demand.

In CUCB-Avg and CUCB, we set $\alpha = 2.5$. In Thompson sampling, the prior distribution of p is Unif$[0, 1]^n$.

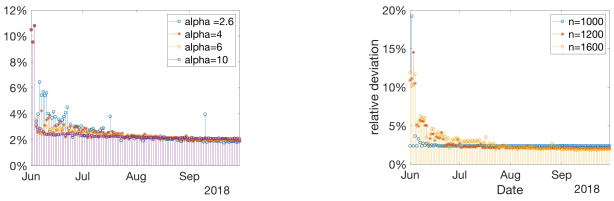
Figure 1 plots the regret of CUCB, CUCB-Avg and TS under two schemes of peak shaving. The x-axis is in log scale. Both figures show that CUCB-Avg performs better than CUCB and TS. In addition, the regret of CUCB-Avg in Figure 1(a) is linear with respect to $\log(T)$, consistent with our theoretical result in Theorem 2. Moreover, the regret of CUCB-Avg in Figure 1(b) is almost linear with $\log(T)$, demonstrating that in practice the regret can be much better than our worst case regret bound in Theorem 4.

Figure 2 plots the 90% confidence interval of the relative reduction error, $\frac{\sum_{i \in S_t} X_{t,i} - D_t}{D_t}$, of CUCB-Avg and TS by 1000 simulations. It is observed that the relative error of CUCB-Avg roughly stays within ±5%, much better than Thompson sampling. This again demonstrates the reliability of CUCB-Avg. Interestingly, the figure shows that TS tends to reduce less load than the target, which is possibly because TS overestimates the customers' load reduction when selecting customers. Finally, on August 18th both algorithms cannot fulfill the daily peak target because it is very hot and the target is very large.

6.2 Effect of α and n

Next, we discuss the effect of the tradeoff parameter α by comparing the relative deviation of the load reduction, $\sqrt{\mathbb{E}\, L(S_t)}/D_t$, of CUCB-Avg with different α in Figure 3(a). The target is determined by scheme (i). It is observed that when T is small, a large α provides smaller relative deviation. This is because when T is small, the information of customers is limited. A larger α encourages exploration of the information, yielding better performance. When T is large, a smaller α leads to a better performance. This is because when T is large, the information of customers is sufficient, and a small α encourages the exploitation of the current information, thus generating better decisions. In addition, Figure 3(a) shows that for a wide range of α's values, CUCB-Avg reduces the deviation to below 5% after a few days, showing that CUCB-Avg is pretty robust to the choice of α.

Finally, we discuss the effect of the total number of the customers n. The target is determined by scheme (i). Figure 3(b) shows that even with a large number of customers, CUCB-Avg reduces the relative deviation to below 5% very quickly, demonstrating that our algorithm can handle large n effectively. In addition, when T is small, a small n provides smaller relative deviation, because a small number of customers is easier to learn in a short time period. When T is large, a large n provides better performance, because there are more reliable customers to choose from a larger customer pool.



(a) Effect of α (b) Effect of n

Fig. 3. The standard deviation of the actual load reduction of CUCB-Avg with different α or different number of arms n.

7 Conclusion

This paper studies a CMAB problem motivated by residential demand response with the goal of minimizing the difference between the total load adjustment and the target value. We propose CUCB-Avg, and show that CUCB-Avg achieves sublinear regrets in both static and time-varying cases. Future work includes considering dynamic population and general customer behavior models, e.g. continuous distributions, Markov processes, etc.

References

[1] Pierluigi Siano. Demand response and smart grid – a survey. *Renewable and Sustainable Energy Reviews*, 30(C):461–478, 2014.

[2] Na Li, Lijun Chen, and Steven H Low. Optimal demand response based on utility maximization in power networks. In *2011 IEEE power and energy society general meeting*, pages 1–8. IEEE, 2011.

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

[4] NYISO. New york ISO demand response program. `http://www.nyiso.com/public\\/markets\_operations/market\_data/demand\_response/index.jsp`.

[5] Farrokh Rahimi and Ali Ipakchi. Demand response as a market resource under the smart grid paradigm. *IEEE Transactions on Smart Grid*, 1(1):82–88, 2010.

[6] FERC. Reports on Demand Response and Advanced Metering . Technical report, Federal Energy Regulatory Commission, 12 2017.

[7] Daniel O'Neill, Marco Levorato, Andrea Goldsmith, and Urbashi Mitra. Residential demand response using reinforcement learning. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 409–414. IEEE, 2010.

[8] Haiwang Zhong, Le Xie, and Qing Xia. Coupon incentive-based demand response: Theory and case study. *IEEE Transactions on Power Systems*, 28(2):1266–1276, 2013.

[9] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[10] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[11] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *The Journal of Machine Learning Research*, 17(1):1746–1778, 2016.

[12] Richard Combes, Mohammad Sadegh Talebi Mazraeh Shahi, Alexandre Proutiere, et al. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems*, pages 2116–2124, 2015.

[13] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking (TON)*, 20(5):1466–1478, 2012.

[14] Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson. Matroid bandits: Fast combinatorial optimization with learning. *arXiv preprint arXiv:1403.5045*, 2014.

[15] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, pages 535–543, 2015.

[16] Qingsi Wang, Mingyan Liu, and Johanna L Mathieu. Adaptive demand response: Online learning of restless and controlled bandits. In *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*, pages 752–757. IEEE, 2014.

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

[18] Shweta Jain, Balakrishnan Narayanaswamy, and Y Narahari. A multiarmed bandit incentive mechanism for crowdsourcing demand response in smart grids. In *AAAI*, pages 721–727, 2014.

[19] Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, and Ian Osband. A tutorial on thompson sampling. *arXiv preprint arXiv:1707.02038*, 2017.

[20] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–1, 2012.

[21] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011.

[22] Con Edison. Consolidated edison smart ac program. `https://conedsmartac.com`.

[23] Jean-Yves Audibert, Sébastien Bubeck, and Gábor Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39(1):31–45, 2013.

[24] Aditya Gopalan, Shie Mannor, and Yishay Mansour. Thompson sampling for complex online problems. In *International Conference on Machine Learning*, pages 100–108, 2014.

[25] Siwei Wang and Wei Chen. Thompson sampling for combinatorial semi-bandits. In *International Conference on Machine Learning*, pages 5101–5109, 2018.

[26] Yingying Li, Qinran Hu, and Na Li. Learning and selecting the right customers for reliability: A multi-armed bandit approach. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4869–4874. IEEE, 2018.

[27] Yingying Li, Qinran Hu, and Na Li. Learning and selecting users for achieving reliability: A multi-armed bandit approach. 2019. online report, `https://nali.seas.harvard.edu/files/nali/files/automaticadr.pdf`.

[28] NEISO. New england iso express load and demand data. `https://www.iso-ne.com/isoexpress/web/reports/load-and-demand/-/tree/demand-by-zone`.